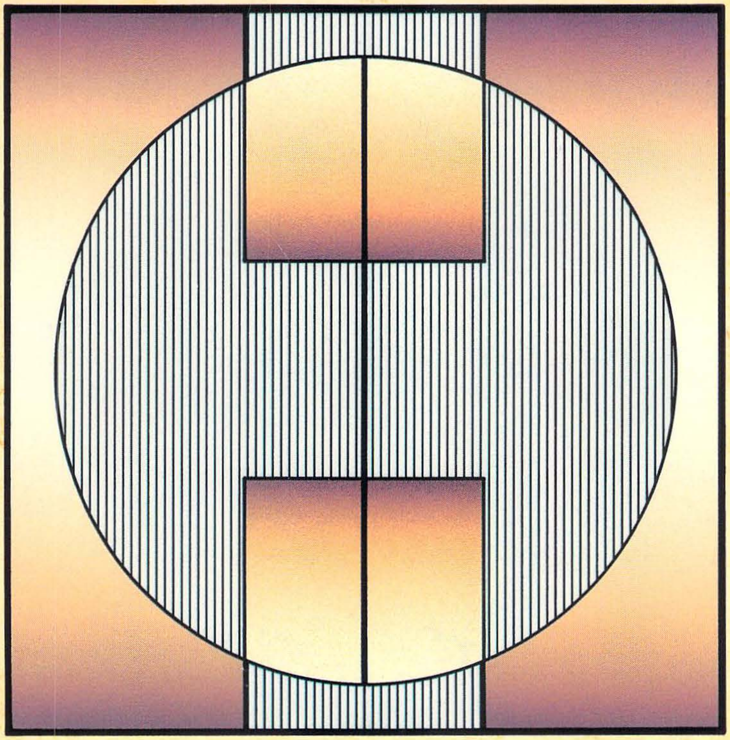
HARD DISK MANAGEMENT



Stephen Fisher and Lynn Frantz

File management, partitions, directories, and much more. The complete how-to reference for any MS-DOS user who wants to get the most from a hard disk.

COMPUTE! LIBRARY SELECTION

HARD DISK MANAGEMENT

Stephen Fisher and Lynn Frantz

COMPUTE! Publications, Inc. abc
A Capital Cities/ABC, Inc. Company
Greensboro, North Carolina

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-87455-116-1

COMPUTE! Publications, Inc., Post Office Box 5406, Greensboro, NC 27403, (919) 275-9809, is a Capital Cities/ABC, Inc. Company and is not associated with any manufacturer of personal computers. IBM PC, PC XT, and PC AT are trademarks of International Business Machines, Inc.

Contents

Foreword

Here's the complete hard disk companion—a book that covers all phases of hard disk management, from making the correct first decisions through installation to damage control.

In these pages, you will find descriptions of the hard disk alternatives, including internally and externally mounted units, hard disks on plug-in boards, and the new alternatives in mass storage like the Bernoulli Box, VCRs, and optical media. There's also a thorough discussion of the factors that might weigh in your decision whether or not to purchase one of these alternative devices—not based on the first hour or day of use, but from a perspective farther down the road. With 30, 60, or 100 megabytes of storage at your disposal, it's difficult to imagine a day when you will run out of room. Yet, without the proper planning and preparation, even the fastest and most spacious hard disk may become slow, cramped, and unreliable. This book takes the long view, and that can make all the difference.

Turn to Chapter 2 to learn what applications and data files belong on the hard disk—and which are better left on floppy disks. Chapter 3 will tell you everything a hard disk owner needs to know about hard disk installation and maintenance, and how to structure a trouble-free physical environment for your equipment that will keep maintenance to a minimum.

Chapter 6 reveals ways to streamline your system with hardware and software to provide the maximum in speed and storage—it doesn't cost as much as you might suspect.

Security, encryption, backup, and utilities packages are all discussed in this easy-to-read, yet authoritative text.

Chapter 1

Technology and Impact

Technology and Impact

The computer story is a story of power and speed. All efforts in the field are directed at doing more, faster. How much processing can a computer do, how quickly? Tests for speed are straightforward. The tests are:

- How fast can the computer retrieve information?
- How fast can it do critical operations?

Capacity is slightly more complex. The key measures are:

- Internal memory
- External storage capacity
- Processing sophistication

Computers need memory to hold programs and data. Internal or random access memory (RAM) is fast, volatile, and relatively costly. It is *volatile* because memory is only preserved as long as the power is on. When you turn off the power you will lose the information stored in RAM.

Read only memory, or ROM, is not as fast as RAM, and its contents never change. ROM typically holds the computer's *firmware,* which controls loading the operating system, or the operating system itself.

Computers traditionally use limited amounts of RAM and ROM. All other supporting memory is slower, cheaper external memory: tape drives, floppy disks, and hard disks. RAM is usually measured in kilobytes. One kilobyte, written *1K,* is 1024 characters. In the early 1980s, most microcomputers were 64K machines with 180K floppy drives.

As capacities increase, so do units of measure. Today, it is not unusual to find machines with a megabyte (1024K, written 1Mb) of RAM, although a typical new machine might have 512K or 640K. These machines often come equipped with a hard disk that has a capacity of between 10Mb and 30Mb. Some hard disks now have

capacities ranging into the gigabytes (thousands of megabytes). Many of today's machines are also much faster than the first PCs.

Along with greater speed and capacity has come greater sophistication. Programs are now larger and more complex than those introduced ten years ago. New features and functions range from grammar checkers and English-like command interpreters to multilevel database and financial analysis programs. Operating system advances allow data sharing, memory sharing, and program switching.

Why Hard Disks?

In this high-speed, high-power environment, hard disks make sense. Hard disks increase both speed and capacity. The slowest hard disks are about ten times faster than the fastest floppy disk drives. The smallest hold nearly 30 times as much data.

Buying a hard disk is buying efficiency. Here are a few ways capacity translates into convenience:

- Larger programs. Many of today's most useful programs are big. Often they require 512K RAM to run efficiently. Even with 640K, the whole program may not fit into RAM at one time. These large, complex programs far exceed a floppy disk's capacity. The best-selling word processors, database managers and spreadsheets include five or six program disks. Using all the features of these programs with a floppy disk drive would mean frequent disk switching.
- Large data files. As programs have grown, so have data files. Extensive accounting, database, or word processing files may not even fit on a single disk. Without a hard disk, you'd have to split files arbitrarily into smaller, more manageable chunks.
- Sharing data between programs. One sign of greater sophistication is that programs can work with the same data in different ways. You may enter your data in one program, analyze it in another, graph it in a third and create a finished document in a fourth. Accomplishing this with floppy disks would be a tremendous chore.
- Switching between programs. Another advance in operating systems is the ability to suspend a program in the midst of operation,

use another, then return to the first. As above, without the support of a hard disk, this would be a time-consuming task.

But Can I Afford It?

Hard disk prices have dropped drastically in the last few years. This is one reason they're so common. Drives that are 10Mb or 20Mb start around $300. A hard disk unit may cost only three times as much as a floppy unit but provide fifty times more storage. Hard disks cost no more than tape units and are much faster and more flexible.

Hard disk costs have also dropped in another way: Today's drives require less space and less power than early models. Smaller, lighter units fit easily inside your computer and demand less from your system's power supply.

Physical Design

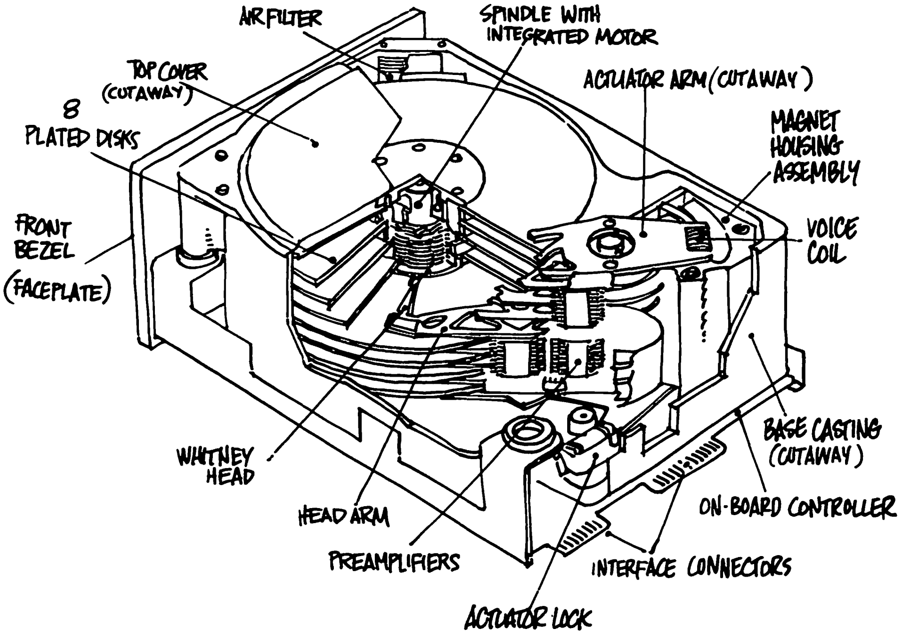
A hard disk system is a computer storage unit using rigid disks. However, a unit is more than just a hard disk. As Figure 1-1 illustrates, each unit or drive contains

- One or more platters or disk surfaces
- A read-write head and supporting arm for each platter
- A special motor to move the arms and heads
- A central spindle or shaft
- A drive motor which turns the spindle
- Electronic control circuitry

You may hear several names for hard disks. Hard disks can be either removable or nonremovable. Most are nonremovable: The platters are sealed inside the unit. IBM calls nonremovable hard disk drives *fixed disks*. Some people call them *rigid disks*. They are also commonly referred to as *Winchester drives*.

In the early 1970's, IBM was working in secret to improve disk drive technology. Their 3330 removable cartridge drives had reached the limits of the *flying head* technology. If they spun the

Figure 1-1. Anatomy of a Hard Disk Drive



disk faster or moved the head closer to the recording surface, the airflow created too much lift. If they increased the downward pressure on the head, minor air turbulence or surface grit would cause the head to plow into the disk drive. IBM engineers decided to hermetically seal the drive platters, spindle, and heads to reduce both airflow and contamination. The prototype unit was a combination of an existing 30Mb removable cartridge drive with another 30Mb nonremovable, hermetically sealed drive. This 30-30 design project was code-named *Winchester*. The IBM-watching trade press made the secret name synonymous with sealed nonremovable hard disk technology.

The Disk Platters

Hard disk platters are usually rigid aluminum coated with a magnetic metal oxide. This metal oxide is similar to that on floppy disks or tapes. Magnetic particles rest in a nonmagnetic "glue."

In newer *thin-film media* units, the metal oxide layer bonds directly to the disk surface. This magnetic layer is more compact,

roughly one-tenth the thickness of traditional oxides. The entire surface—not just occasional particles—is magnetic, so the same information can be recorded in a much smaller space. Furthermore, the metal coating is harder than most read-write heads, so it's much more resistant to impact damage.

Most microcomputer hard disk units have two platters per drive, but there may be as many as eight. Platters are usually 3½ or 5¼ inches in diameter; larger sizes are available. Capacity varies widely for any size. Most 3½-inch drives hold 10Mb to 30Mb; 5¼-inch units hold from 10Mb to 40Mb.

Read-Write Heads, Arms, and Motors

Hard disk drives store information on both sides of each platter. Each usable side has its own read-write head. The head records data by changing magnetic patterns on the disk. It reads data by sensing the magnetic patterns already present. Each read-write head has its own arm. The arm supports and moves the head, just as a record player arm supports and moves a needle.

All arms move at once, under control of a special motor called a *head actuator.* The arms move very quickly in small, precise steps. This speed and precision are the main differences between hard disk arms and those used with floppy disks.

There are two main methods of controlling head movement. *Open loop* systems go directly to an expected location with no feedback on actual location. Most open loop systems use *band-stepper* technology. A stepper motor moves the arms one track each time it sends a pulse. Band-stepper systems tend to be slow because they can't reliably send and receive pulses beyond a certain speed.

Closed loop systems check as they go to make sure they end up in the right place. The most common checking device is a dedicated surface, called a *servo surface,* containing special location identifiers. A voice-coil solenoid, similar to the voice coil in a loudspeaker, pulls the arms out from a spring. The heads move quickly to the approximate location, then send back location data enabling them to finish their journey to the right place.

Because they don't have to use discrete pulses, servo-voice coil systems are faster than open-loop systems. However, an entire surface is lost to the feedback system, so closed-loop systems are more expensive. Also, servo-voice coil systems are often noisy and jerky because of the energy required to move the arms.

The Spindle and Drive Motor

All platters connect to a central rotating shaft or spindle. You might compare it to the spindle on a record player. However, a record player spindle only aligns records; it doesn't actually turn them. A hard disk spindle, by contrast, does the work of spinning the platters. A drive motor provides the power to turn the spindle. The motor connects to the spindle either with a belt-and-pulley system (belt-drive) or directly (direct-drive). Most new drives use direct-drive connections.

Hard disks normally spin at 3600 RPM. Unlike floppy disks, they spin constantly, because the heavier hard disks are hard to stop and start. The extra energy required to maintain constant spin means that hard disks draw much more power than floppies.

Electronic Control Circuitry

Complex electronic circuitry controls the drive. The circuits receive commands from the operating system, translate them, and issue appropriate directives moving the heads, reading or writing data, and so on. Although rare, circuitry problems cause havoc when they occur. The unit writes data to the wrong locations and reads the wrong data. Major damage can happen in a short time.

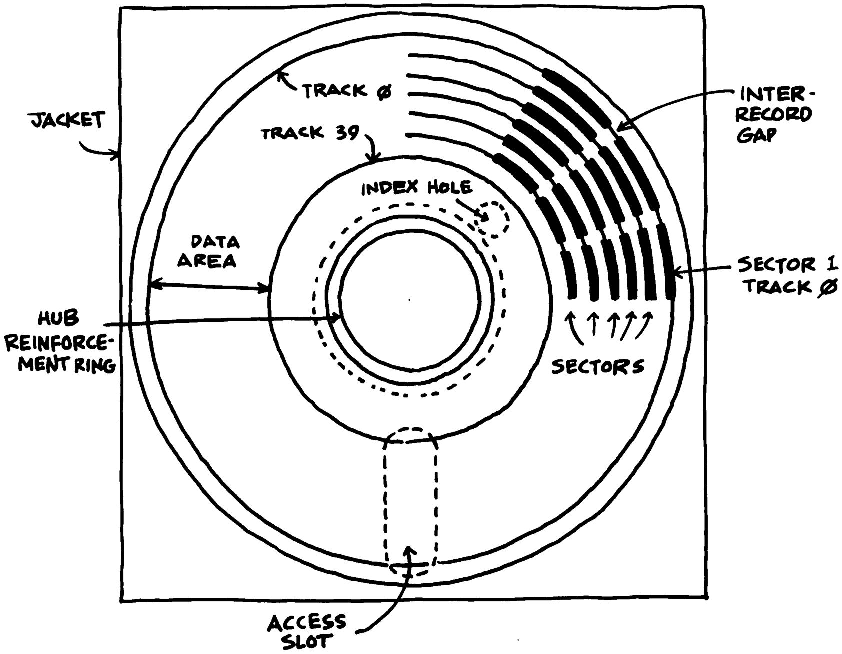
How Hard Disks Differ from Floppy Disks

Hard disks are similar to floppy disks in many ways. Both use spinning disks coated with a special magnetic metal oxide. Both use read-write heads to rearrange magnetic patterns on the disks and then sense what was written. Indeed, all the major components of hard disks are also present in floppy disk systems.

Many of the differences between hard disks and floppy disks are differences of degree. Nonetheless, the result is that the hard disk is a vastly more powerful storage unit. Therefore, here is a summary of the major advances between floppy disks and hard disks. Compared to floppy disks, hard disks have:

- An inflexible metal base instead of a flexible plastic base
- Thinner, more compact magnetic metal oxide coating
- A faster rotation speed for faster data transfer
- Constant rotation for instant availability
- Higher power requirements

Figure 1-2. Logical Topography of a Soft-Sectored Floppy Disk



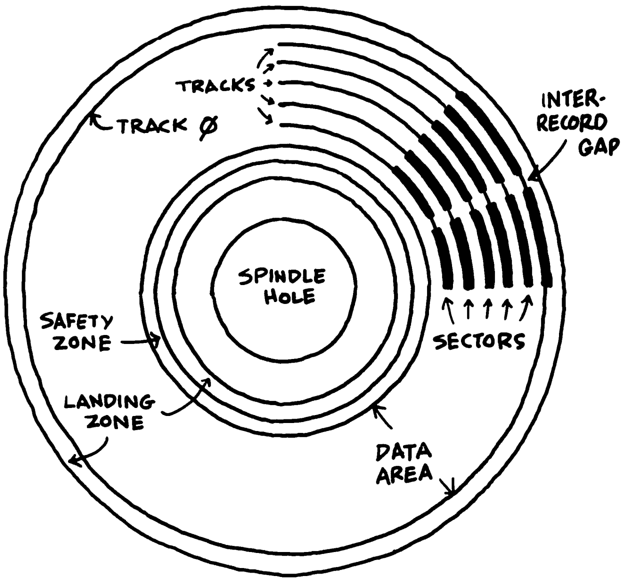
- Faster track-to-track head positioning
- A flying head (results in no media wear)
- A narrower head (produces more tracks per inch)
- Advanced circuitry for denser data recording

However, the distinctions between floppy and hard disks are blurring. Iomega's Bernoulli Box and Kodak's Verbatim Internal Subsystem combine a traditional flexible disk with tighter stepping control and a more accurate head. The head sits right on the disk, like floppy disks. Their rotation speeds fall between floppy disks and hard disks. Their overall data transfer speed is equal to slow hard disks, much faster than floppy disks. Like other removable media systems, they combine speed with versatility and data security.

The Logical Structure of Hard Disks

When early settlers came to the midwestern United States, they found a vast plain with few outstanding landmarks. They divided the land logically into sections, counties, and later states. Likewise,

Figure 1-3. Logical Topography of a Hard Disk Platter



a hard disk surface is an undifferentiated metal oxide plain. To know where to read and write, the disk controller logically identifies each small piece of the surface area. The units of the disk landscape are *sectors, clusters, tracks,* and *cylinders.*

Sectors, Tracks, and Cylinders. Sectors are the smallest logical building blocks of both floppy disks and hard disks. A sector is a section of disk large enough to store a defined number of characters (bytes) of information. Some controllers manage to squeeze up to 1024 bytes into each sector, while some put only 512 or 128 bytes in a sector.

On each recording surface, sectors sit in concentric circles called tracks. Track size depends on how tightly information can be packed onto a disk. Three measures affect track spacing:

- Recording density of the magnetic medium
- Head size
- Smallest controlled arm movement

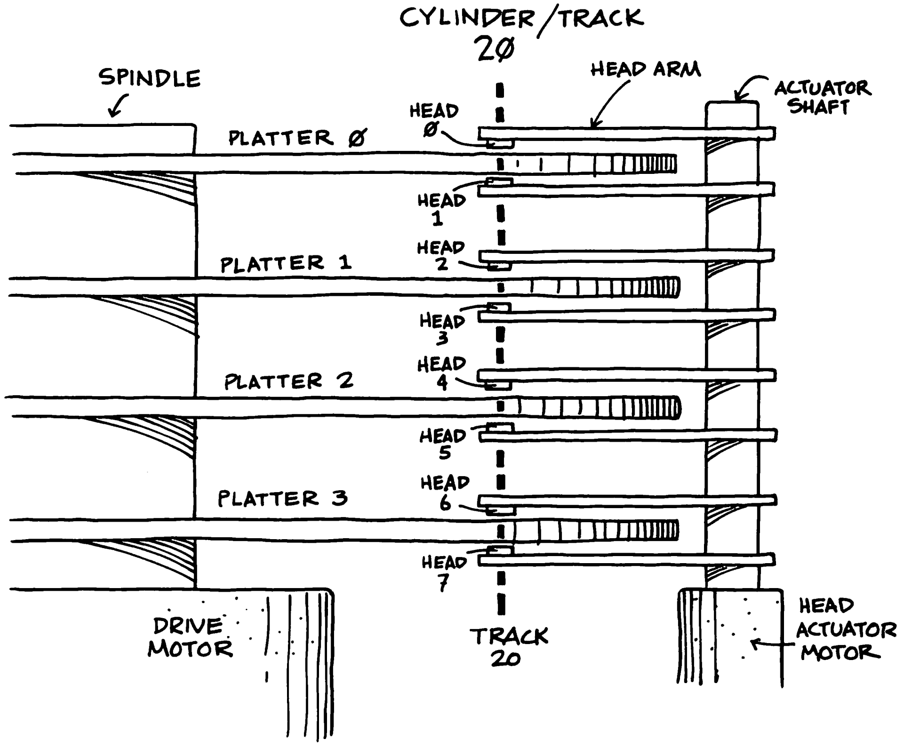
Smaller heads, smaller steps, and higher density media all allow more tracks per surface area. Most floppy disks have only 40 to 80 tracks per surface. Most hard disks have between 300 and 1024.

Inner tracks are much smaller than outer ones. Therefore, the DOS could theoretically put more sectors on outer tracks. However, keeping track of locations would become ridiculously complex. Instead, MS-DOS assigns a standard number of sectors per track. Most floppy disks use nine or 15 sectors per track. IBM hard disk drives usually use 17.

Remember that hard disks usually have at least two platters (four surfaces). Since all heads move together, they are always positioned over identical tracks on different surfaces. A geometric solid connecting those tracks would form a cylinder. Thus, a set of matching tracks from all surfaces of a hard disk is called a *cylinder*. Figure 1-4 depicts such a cylinder.

Figure 1-4. Side View of Hard Disk

The read-write heads move in unison, forming a cylinder of tracks that are simultaneously available.



MS-DOS numbers all sectors sequentially, one cylinder at a time. The first cylinder, cylinder zero, is the outermost set of tracks. Likewise, the first platter is platter zero, and the first surface of each platter is called side zero. Depending upon the brand of disk controller chip, the first sector on cylinder zero, platter zero, side zero is called either sector zero or sector one.

After using all the sectors on the first side, MS-DOS moves to side one (the second side) of cylinder zero, platter zero. It next moves to side zero, platter one. After finishing both sides of platter one, it moves to the next platter and repeats the process. It continues to move from platter to platter until it completes cylinder zero. Then it starts on cylinder one.

MS-DOS fills a cylinder at a time, rather than a platter or a side for the sake of efficiency. Remember, all heads line up on a single cylinder at a time. They can read or write an entire cylinder's data without moving. Switching heads is roughly one hundred times faster than moving heads, so cylinder-by-cylinder processing yields the fastest possible reads and writes.

Clusters. MS-DOS has one more way of organizing information on the hard disk: by clusters. A cluster is the smallest unit MS-DOS uses to read and write files. Cluster size depends on the storage unit and version of MS-DOS. It ranges from one to 16 sectors (usually 512 to 8096 bytes). MS-DOS versions 3.0 or greater use eight-sector clusters on 10Mb disks and four-sector clusters on larger disks.

Every file, no matter how small, uses at least one full cluster. If a file is one byte longer than a cluster, it will occupy an entire extra cluster.

Cylinder zero. Cylinder zero is special. Side one, cylinder zero, sector one, always contains the master boot record and master partition table for a disk.

One physical disk can split into as many as four logical partitions. Each partition can run a different operating system if desired. The master partition table describes each partition:

- Whether this is the active partition
- Starting location: cylinder, head, and sector
- Operating system
- Ending location
- Number of sectors before the start of the partition
- Number of sectors in the partition

Only one partition can be in charge at a time. The partition in charge when the system *boots* (starts) is called the *active partition.* The master boot record contains a two-step startup procedure.

- Search the master partition table for the active partition.
- Load the first sector of the active partition. This should be a startup (boot) record for the operating system controlling this partition.

If the disk is entirely a DOS disk, the MS-DOS boot record is in sector two of cylinder zero, followed by several system files discussed later in this chapter:

- Two copies of the File Allocation Table
- The root directory
- The MS-DOS BIOS

The MS-DOS boot record contains all information needed to start MS-DOS, including:

- Which version of MS-DOS formatted the disk
- The names of hidden MS-DOS files
- Instructions to find and load the hidden MS-DOS files
- The sector size, in bytes
- The cluster and cylinder size, in sectors
- Number of cylinders on the disk
- File Allocation Table size
- Root directory size

The File Allocation Table

The file allocation table (FAT) records the status of every sector on a disk. Any sector is available for use, is bad, or is part of a file. Hence, FAT entries track each sector in one of three ways:

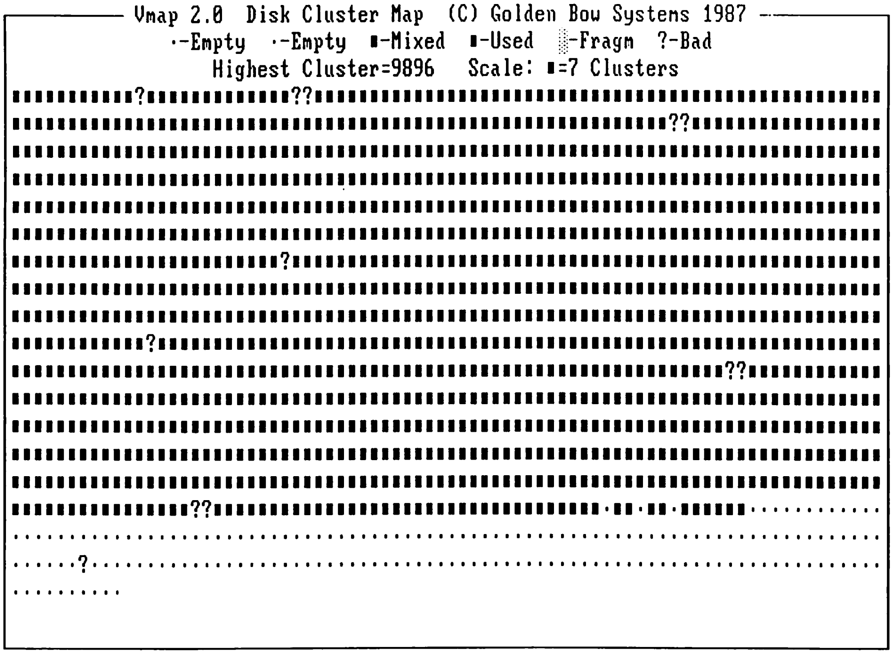
- Free sectors marked as available (unused);
- Bad sectors marked as such (unused but unavailable);
- Sectors in files marked with the address of the next sector in the same file.

The third option is important. It means the sector entries for one file form a linked list. If you know where to start, you can follow the trail from sector to sector until you find all the pieces of the file. Hence, the main job of the file allocation table is keeping

track of file locations. Figure 1-5 shows the screen display of a program that translates the file allocation table to a visual rendition of disk usage.

Figure 1-5. Screen Display of *Vopt* VMAP Program

The Vopt *VMAP displays the disk usage as shown in the file allocation table.*



You probably recall that MS-DOS doesn't read or write files in units smaller than clusters. This carries over to the file allocation table. Although each sector has its own record-keeping, files grow and shrink one cluster at a time. If one sector is added to a file, all other sectors in its cluster join the file's linked list.

The file allocation table changes more frequently than anything else on the disk. It also holds the most crucial information for proper use of the disk. Therefore, MS-DOS keeps two identical copies of the FAT. The two are next to each other in cylinder zero.

The Root Directory

The file allocation table tells where a file is. It doesn't tell its name, when it was created, or anything else about the file. Directories do that job. As you can see in the screen display of Figure 1-6, your disk may have many levels of directories. Directories and subdirectories will be discussed at length in Chapter 4. The focus of this discussion will be on the root directory, which is the only required directory and the only directory on a newly formatted disk.

Figure 1-6. Disk Directory Structure

The public-domain VTREE *program shows the tree-like disk directory structure.*

```
D:\>vtree
┬─ PUB ──────────┬─ ARTWORK
│                ├─ DATA
│                ├─ DOTFONTS
│                └─ MACFONTS
├─ CIELOGO
├─ COMM ─────────┬─ KERMIT ──────┬─ MSKERMIT
│                │               └─ SOURCE
│                ├─ MIRROR
│                ├─ MOVE-IT
│                ├─ PROCOMM
│                └─ XTALK
├─ DISKBOOK
├─ GMK
├─ SN ──────────── DEMO
├─ TYPESET
├─ VENTURA
├─ WINDOWS ─────── PIF
├─ WORD ─────────┬─ KBLEARN
│                └─ MSLEARN
├─ WP ──────────── DATA
└─ WS4 ──────────┬─ LETTERS
                 └─ TUTORIAL
```

Unlike other directories, the root directory is fixed in size. All root directory sectors are assigned when the disk is first formatted. Each directory entry is 32 bytes long, so a 512-byte sector holds 16 directory entries. The size of the root directory varies with disk size. Floppy disks have from 62 to 224 entries in the root directory. Hard disks have more, but rarely more than 512 entries.

DOS Files

The last set of required files on cylinder zero are the hidden MS-DOS files. For the disk to be bootable, these must go immediately after the root directory. Furthermore, they must be in one unbroken span of adjacent clusters. There are two hidden files:

- The Basic Input/Output System (BIOS) controls hardware interactions. Each manufacturer's equipment has its own BIOS with appropriate low-level hardware commands. The BIOS file is IBMBIO.COM on IBM's PC-DOS; elsewhere it may be IO.SYS, BIO.SYS, or something else.
- The Basic Disk Operating System (BDOS) is the heart of the operating system. BDOS is the same on all systems. This is what allows programs to run on many brands of computers. MSDOS.SYS is the filename in the generic version; on IBM systems it is IBMDOS.COM.

Summary

This chapter has telescoped the development of hard disks, explained their usefulness, and outlined their physical and logical design. If you're ready to move into the world of hard disks, turn to Chapter 2 and learn how to pick the unit that's best for you.

Chapter 2

Hard Disk Selection

Hard Disk Selection

This chapter examines the features and tradeoffs of the various types of hard disk storage. It explains the factors in choosing a hard disk, helps you analyze how they apply, and shows how to weigh their importance. Read this chapter if:

- You're still planning your entire system.
- You're upgrading from floppy disks to a hard disk.
- You already have a hard disk and want either to replace it or to add another.
- You're curious about decisions you've already made.

When selecting a hard disk, the factors to consider are

- Desk space
- Disk capacity
- Reliability
- Removability
- Speed
- Compatibility
- Ease of installation
- Maintenance
- Software support
- Future expansion
- Price and where to buy

How Much Hard Disk Storage Is Enough?

The logical first decision you must make is what size hard disk you need. To decide that, you must first decide how you'll use the drive—where you'll store your programs and your data. You have three basic options:

- Put everything on a nonremovable hard disk
- Load only programs and static data on the hard disk
- Use removable media, such as disk cartridges

Here is the strategy behind hard disk shopping:

- Know what you need before you go shopping. Don't let a salesperson talk you into anything else. Buy what you need, not what is on sale.
- Determine your storage requirements.
- When buying a system, consider a dual-floppy computer to which you add the hard disk that best suits you.
- If you can choose between an internally-mounted and an externally-housed drive, pick the internal drive. Upgrading your computer's power supply saves money and space over buying an external cabinet with its own power supply.
- If you must add a controller card with the hard disk, consider getting a disk-on-a-card (known as a *hard disk card*) instead. They are standard, fast, low-power, easy to install, and moderately priced. Some can support a conventional drive as their second hard disk. The IBM PC AT and its clones already include a hard disk controller, so most hard disk cards may only function as secondary drives in these systems.
- Replace a floppy drive with a hard disk only as a last resort. Having two floppy disk drives is handy. You can use them for backups and disk copying, or you can have two different capacity drives. The Panasonic JU-4752-AEG is a half-height 5¼-inch, high-density, 1.2Mb drive that formats and writes 360KB low-density disks that are consistently readable by standard PC and PC XT low-density drives. Disks created on regular AT-style high-density drives are often unreadable by older PCs.
- When upgrading a dual-floppy PC, consider replacing a single full-height floppy disk drive with dual half-height drives. You

Most hard disk users keep all their programs and data on one or two nonremovable hard drives. This is the fastest and easiest way to use a computer, but it needs the most disk space. To protect your data, you'll need to perform frequent backups (see Chapter 8).

Another method is to keep all your programs and system files on a hard disk, using floppy disks for your data. This is especially useful when many people share a pool of computers. When every machine contains the same programs, they become interchangeable *virtual software machines* or *generic software machines*. This option reduces hard disk storage, reduces the need for frequent hard-disk backups, keeps data private, and (in the case of multiple machines)

can then put either a single full-height or dual half-height hard disk in the other floppy disk bay.

- Stick to industry standards to insure compatibility with the widest range of enhancement products. Avoid special device drivers or boot disks.
- Decide if you need special features such as extra capacity, removable media, or shock mounting. Always get a drive that automatically moves the read/write heads away from the data area when you turn it off. This automatic *park and lock* prevents the damage arising from inadvertent bumps. Protect against the vacuum cleaners of the late-night janitorial staff.
- If you select a large-capacity drive (in excess of the 32Mb supported by MS-DOS 3.2 or earlier), consider upgrading to MS-DOS 3.3 to avoid using special device drivers.
- Divide large-capacity drives (over 32Mb) into smaller logical drives (20Mb or so) unless an application requires more contiguous storage. MS-DOS finds files faster within small volumes than within large ones. Be aware, however, that few database programs support a single file that spans multiple drives.
- Select a drive that is faster than you need at this time. A PC XT with a standard 4.77MHz 8086 processor works fine using a hard disk with a 65ms (millisecond) step rate. A standard AT with 6MHz or 8MHz 80286 processor zips along with a 40ms drive. If you upgrade the motherboard (for about $500) to a 10MHz 80286 or (for about $1,500) to a 20MHz 80386, you'll wish you had purchased 25ms hard disk drives.
- When you first install your hard disk, use the proper interleave (Chapter 3) and subdirectories (Chapter 4) to make it respond more quickly.

provides on-site hardware redundancy. Individual file sizes are limited to disk capacity, so this is not appropriate for large database applications.

Data transfer using a hard disk is about ten times faster than with floppy disks. System and program files resident on the hard disk will have fast access. These files are read more frequently than any single data file. Copying heavily-used data files either to the hard disk or to a *virtual disk* (a ramdisk where computer memory simulates a hard disk) reduces disk slow-down. But remember to copy the results back to the floppy disk.

Using a hard disk with removable media combines the speed

and capacity of a hard disk with the switching power of floppy disks. However, removable units cost more than standard drives. You probably won't choose this option unless you want removable disks for backup or have one or more of these special needs:

- Extremely large storage requirements, in excess of 50Mb
- More than one operating environment, such as UNIX and MS-DOS or GEM and Windows
- Security requirements that necessitate locking up your data when not in use

Your choice will depend in part on the quantity of data you need to store. You'll need to figure out your storage volume for

- System files and utility/support programs
- Application programs
- Data files for your various applications
- Developmental space for any new programs/applications in the future

When figuring your hard disk storage needs, remember:

- A typewritten page ranges between 3,000 and 4,000 characters (3K to 4K).
- A standard 5¼-inch disk formatted on an IBM PC or compatible holds about 360,000 characters (360K), or 90 pages.
- A high-density PC AT floppy disk holds about 1,200,000 characters (1.2Mb), or 300 pages.
- Four standard floppy disks equal one high-density floppy.

An accurate count must include all your files. If you want on-line backup copies of word processing, spreadsheet, or other data files, allow room for them. If you will be developing your own programs, leave room for compilers and developmental tools as well as all source and object files.

Hard disk management brings its own new storage costs: extra space for subdirectories, new hard disk utilities, and so on. Allow at least 1Mb for these items. No matter how little space you think you need, you should probably purchase a hard disk drive with at least a 20Mb of storage. You'll be amazed at how quickly it will fill up.

Most people can get by with 20Mb to 40Mb of hard disk storage. If you fall in this range, you're in the mainstream and will find many systems available to choose from in all prices, sizes, and

types. If you need more storage than this, study Table 2-1 for examples of high-capacity drives on the market. All are fast. Their average access time is typically between 18 and 30 milliseconds. A representative WORM (Write-Once-Read-Many) drive was included for comparison.

Check Your Power Supply and Open Space

To decide accurately what's best for you, you must know four numbers for your current hardware:

- How much power does your system currently provide? If you have an IBM PC or compatible, the answer is probably 63.5 watts; if you have an XT, 135 watts; an AT, 210. If you've upgraded your power supply, you're probably aware of your current wattage. If you're unsure, check your system manual, call your vendor, or open the machine and look.
- How much extra power is currently available? ATs have a 7-amp power supply, which is adequate for a single full-height hard disk or two half-height drives plus floppy disk drives. XTs that are heavily loaded with extra graphics cards, modems, controllers, and so on occasionally overextend their power supply. If your machine is a PC with a 63.5-watt power supply, and if you have an extended memory card, an internal modem, and/or a multifunction card, you're probably close to your power limits. If your screen display balloons (swells at the edges) when you first turn on your system, you probably need more power.
- How many disk drive bays are currently unused? You don't have to open your machine to see how many bays are in use. Your current drives are at the front of your machine and are either half-height or full-height. Two half-height drives can fit on top of each other, each a little under two inches tall. A full-height drive is about 3½ inches tall. If your machine is in the AT family, it can hold two full-height drives and one half-height. Most other nonportable machines hold two full-height drives or four half-heights. Portables or laptops may hold fewer. Find out from your vendor, your system manuals, or a personal investigation how many half-height and full-height drives your system can hold when fully loaded.
- How many expansion slots are (and will remain) unused? If you don't know, open your machine and look. Are the available slots

Table 2-1. Comparison of High-Capacity Hard Disks

Manufacturer	Drive	Capacity	List Price	Comment
Standard Hard Disk Drives				
Conner Peripherals	CP3100	100Mb	$1,595	
Alloy Computer Products	Alloy ID-160	125Mb	$5,995	
Core International	Core HC150	150Mb	$4,995	
Emerald Systems Corp.	DOS 150-3000	150Mb	$2,795	Systems available up to 630Mb.
Emulex Corp.	ATS-380	310Mb	$7,105	Includes a built-in 60Mb cartridge tape back-up and ESDI interface.
Priam Systems Division	Priam ID130	130Mb	$3,598	
Storage Dimensions	AT 160F	160Mb	$3,495	
Write-Once, Read-Many Drives (WORMs)				
Information Storage Inc.	ISI 525WC	230Mb	$2,795	
	HD525	1 Gigabyte	$4,495	

long or short? How much could you rearrange the current boards if needed? Be sure to consider future growth. If you plan to add an extended memory card, internal modem, mouse, serial printer connections, graphics boards, tape backup or any other kind of device controllers, keep the needed slots free.

Internal Versus External Disks

Once you know how much hard disk storage you need and how much room and power you have to spare, it's time to decide on features. Your first decision is whether you'll buy an internal hard disk that fits inside your current system, or an external unit that sits on the desktop as a separate piece of equipment.

Most people choose an internal hard disk. It's one less item to move, clean, and keep track of. It keeps your computer's *footprint* (desktop space requirements) from growing. And it usually costs about $150 less than a comparable external unit, because it doesn't need the extra casing and power supply an external unit requires.

Assuming space on your desktop is not at a premium, there are four possible reasons to buck the trend and select an external unit:

- Inadequate power supply
- Lack of internal room
- Special features in external drive
- The fear factor

Inadequate power supply. After having read the preceding section, you probably know whether your power supply can support a new internal disk. If it is inadequate to the task, you can upgrade your power supply for about $75 to $150 if you install it yourself. Another option is to choose an external disk unit and avoid extra demands on your power supply.

Usually less power is required for 3½-inch hard disk drives. Hard disk cards use 3½-inch drives exclusively. They are also used occasionally on internal drives.

Lack of internal room. There are two possible places for an internal hard disk:

- In the disk drive bays (the area containing your current floppy and/or hard drives)
- In your machine's expansion slots

Both floppy and hard disk drives come in half-height as well as full-height models. Thus, if you're willing to replace some of your current hardware, you have room for at least two floppy drives and two hard drives, even on an ordinary PC. Check with your dealer before trying this, because four drives generate more heat than two.

Hard disk cards don't take up any space in your disk drive bays. Rather, they use one or more expansion slots within your computer. They cost about the same as standard disk drives.

Hard disk cards are easy to install, provided that expansion slots are available, and providing that you did your homework before deciding which one to buy.

Special features. Obviously, if the special features you want are only available on external hard disks, your choice is made for you. However, if you are persistent, you might find an internal unit with the features you are looking for. In the past, removability and extremely high capacity were only available in external units. However, both are now available internally. Even units with built-in tape backup are available internally.

The fear factor. Many people buy external drives because they're afraid to install an internal drive. This shouldn't be a concern. Most internal drives are easy to install with only a screwdriver. Many dealers will install a hard disk for a small extra charge. Some will do it free. If fear of opening your computer is preventing you from buying an internal hard disk drive, consider this: Even external drive installation requires opening the machine and making some connections.

You might want to consider buying your drive from a system integrator. Their drives are designed and tested to be fully compatible with their controller and software. Also, they typically provide well-respected, reliable components and extra support in the form of better manuals, free telephone technical support, and extended warranties. Some integrators provide free credit for your current disk system if you decide to upgrade. Finally, their systems are designed for ease of installation.

The value of enlisting a system integrator may be illustrated best by this anecdote: A systems integrator in California had been successfully using Adaptec 2070 RLL disk controllers with Seagate 238 drives. When he switched to the OMTI 5527 controller, he experienced low-level formatting problems. After many fruitless

hours of looking at electrical signals, he decided to try a different host computer. The OMTI/Seagate hardware combination worked fine in a laboratory system with a new high-speed motherboard. After many more hours of poring over schematics, he noticed that the new motherboard included a single-resistor network chip for the data lines where none was present on the old-style motherboard. Adding the 30-cent part solved the problem—parts: $.30; labor: $3,000.00; cost to customer: $.00.

System integrators include IDEAssociates, Mountain Computer, and Sysgen. Perhaps the best-known is Tandon, the only major disk drive manufacturer that does system integration.

Integrated packages tend to be expensive. For example, a 30Mb IDEAssociates system will set you back over one thousand dollars.

Another solution to the fear factor is a well-designed disk on a card. These cost more than traditional drives, but less than integrated systems. Because the integration has already been done in designing the card, you get many of the benefits of the integrated system, but not the extra levels of customer support.

Fixed or Removable

The great majority of hard disks are fixed. They are permanent parts of your computer system and they can't be removed. However, a removable-disk drive may make more sense for you. Here's a list of considerations:

- Is data security extremely important? If so, you'll appreciate being able to remove your data and lock it up at night.
- Do you share your system with several users? Each person can use personal data cartridges.
- Do you move frequently between two machines? Do you use large or numerous data files? If you install identical drives in each machine, switching data files is easy.
- Do you need fast backup? Dual removable drives are among the fastest to back up.

Like hard disk drives in general, removable-disk drives have dropped in price in recent years while adding new features. Late in 1987, for example, Tandon announced a removable 30Mb shock-mounted drive, the Personal Data Pac, for around $350. Unlike

most removable drives, the entire drive, not just the disk, is removable. The drive uses traditional enclosed Winchester technology but is designed to be completely portable. Tandon claims that you can ship it through the mail without special packaging.

Tandon also offers a specially-designed AT-compatible computer, the Pac 286, with slots for two removable Personal Data Pacs. They claim that 30Mb of data can be transfered from one disk to the other in three minutes. If you already have your system, you can buy an Ad-Pac2 external subsystem that holds two Data Pacs for about $500.

The price of the Bernoulli Box (mentioned earlier) has also dropped. The Bernoulli Box is not technically a hard disk, but it falls within the general subject of discussion. Iomega, the distributor, recently cut prices from $3,600 to $2,600 for a 40Mb system while improving speed and performance features.

Removable systems have also been reduced in size. Several systems, including Bernoulli, Sysgen, and Kodak's Verbatim subsystem, fit in half-height internal bays. The Bernoulli Beta 20Mb drive costs about $2,200; a Verbatim 12Mb system is about $1,400.

Some removable drives offer special features. The IDEAssociates Diskit 2 Plus adds built-in hardware encryption/decryption to the security advantages of removable media. A unit with two 10Mb removable-cartridge spindles is priced at $3,595.

Reliability

If you're going to put all your information on one disk, you certainly want that disk to be reliable. As you may be aware, all disks have some potential for damage.

- Individual sectors on the disk can go bad from dirt, dust, wear, or head crashes.
- The drive's motor, belt, or electronic circuitry can fail, making the disk unusable.

In general, you can't improve the reliability of your hard disk, but you can look into these protective features:

- Nonremovable disks are relatively free from outside contamination since the drive unit is sealed. Check reliability especially carefully on removable units.

- Shock-mounting can reduce the probability of a *head crash,* when the drive's read/write head plows a furrow into the data recording area.
- Park and lock technology automatically locks drive heads safely away from your data when not in use. This reduces the likelihood of a head crash.
- Disk utility programs, often provided free with hard disks, can check for bad sectors, mark them as bad, move any affected files, and warn you about possible contamination.
- Some disks are more damage-prone than others. Disks made using the older, less expensive metal oxide technology are more risky than newer thin-film disks.

Compatibility

The world's best hard disk is useless if it doesn't work with your machine. Ask your salesman about compatibility. Even manufacturers may fall prey to compatibility problems. The first IBM AT hard disk system used a Western Digital controller and a CMI drive. It was widely claimed that this hardware combination with PC-DOS caused data corruption. In 1985, IBM stopped using CMI drives, and complaints stopped.

Make sure the drive unit works with the disk controller. By using run-length-limited (RLL) data encoding, disk drives can boost their data capacity by 50 percent, but only certain high-performance drives can provide the more exacting head postioning and recording characteristics necessary for these increased data densities. The new RLL and ERLL technologies increase the chance for compatability problems.

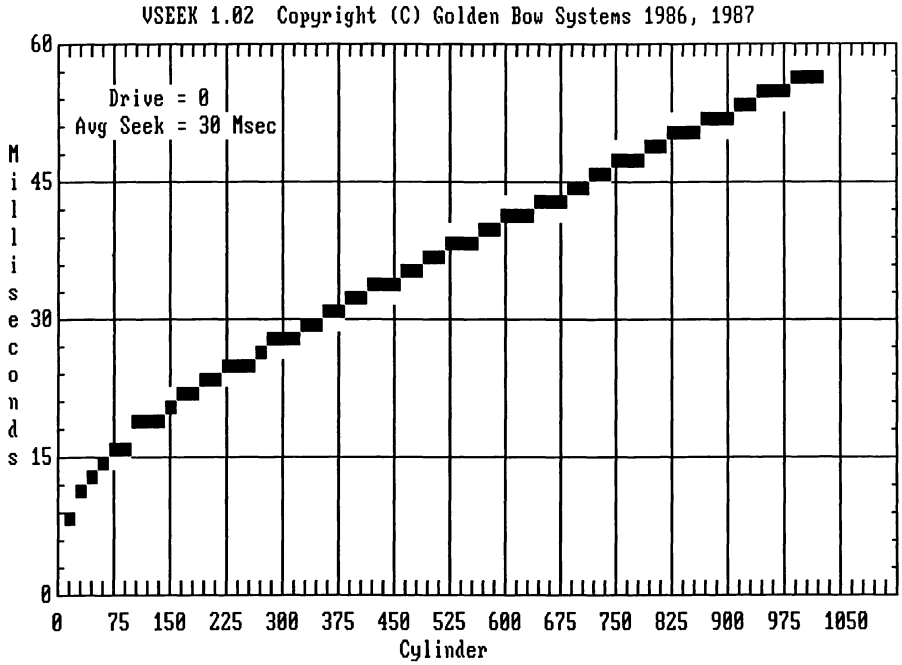
Electronic mail on ARPAnet reported that the Adaptec ACB-1070-A controller for the PC XT does not work in a PC AT unless you replace the IBM BIOS ROM with ROM from a third-party vendor, such as Phoenix Technologies.

If you're considering a second hard disk now or in the future, check carefully for compatibility. Some disk units include a controller that supports a second drive. Some units are neutral. Still others make it impossible to have a second drive. Likewise, some hard drive units cannot function as a second drive. Ask lots of questions and, if possible, arrange a trial run.

Speed

All disk drives that use the same disk controller IC (Integrated Circuit) read and write data at the same speed. They also spin at the same rate. The most useful number for measuring disk performance is average access time. Disk performance measurement has become easier, thanks to the efforts of companies such as Core International and Golden Bow Systems. Figure 2-1 illustrates the performance of a 72Mb Micropolis hard disk on a PC AT.

Figure 2-1. Measuring a Drive's Average Access Time



This measurement is the average time for the head to move from one location to another. Typical times range from 20 to 100ms. Here are a few things to bear in mind about access speed:

- Even the slowest hard disks are nearly ten times faster than floppy disks.
- Prices rise steeply as speeds increase. For less money, you may be able to gain greater speed through cache programs and other techniques discussed in Chapter 6.

- A computer with a slow processor, like the 4.77MHz IBM PC XT, cannot move data fast enough to keep up with a drive stepping faster than 65ms.
- There may not be much correlation between disk speed and your perceived speed as you sit at your machine. If you work mostly on memory-resident documents, spreadsheets, or data files, a faster disk won't help much.

AT (and AT-compatible) owners are the one group with a clear need for speed. ATs are designed to work with fast disks, and they lose much of their speed advantage otherwise. IBM provides hard disks for the AT with an access time of 65ms. If you have an AT, you will probably want a hard disk with an access time of 30ms or less.

Table 2-2 lists some high-speed choices for internal AT-compatible drives. All are 40Mb drives with average access times of 40ms or better, featuring shock mounting and self-parking.

Ease of Installation

Not all hard disks were created equal. Some are very easy to install. Others are not.

Power connections. The simplest hard disks to connect are certain hard disk cards which obtain all their power directly from their expansion slot. Some hard disk cards and all traditional units must be wired to the computer's power supply. If you have a PC with two drives already installed, no spare connector locations are available. You'll have to split one connector into two with a wye (Y) connector.

Also, the short power cables on some hard disks limit where they can be placed. If you're already having trouble squeezing a unit in, this handicap will make your situation even worse.

Size. Traditional drives come in two sizes: half-height and full-height. Hard disk cards aren't so simple. While most claim to fit in one slot, only Plus Development's HardCard, with its custom-designed drive, actually does. The others are too wide. In most positions, they will block the adjacent slot. They are configured to occupy the outside slot. If they don't run into your PC's speaker and if this slot is available, then they only need one slot. The Tandon BusinessCard is such a unit. Still others admit to needing one-and-a-half slots (they work with an adjacent short card). A few

Table 2-2. High-Speed Hard Disks for AT-Compatible Computers

Manufacturer	Drive	Height	Power (in Watts)	Startup Amps	List Price	Comment
Control Data	Wren II	half	20	4.5	$1,335	Requires a special software driver
Core International	AT40F	full	33	4.5	$3,540	Price includes a required additional controller
Microscience	HH-1050	half	15	2.3	$ 850	Noisier than other drives listed
Miniscribe	6053	full	27.50	3.5	$ 819	
Priam	ID-40	full	35	4.5	$1,398	
Seagate Technology	ST251	half	13	2.0	$ 635	Available in quantity only
Tandon	TM755	half	20	2.5	$ 850	$950 with controller

require complicated internal rearranging. One unit requires that you take the speaker off the side of your machine.

Table 2-3 summarizes information on size for a number of popular and well-recommended hard disk cards. All use 10–15 watts of power—less than comparable internal disks.

Documentation and supplementary materials. Are complete, readable installation instructions provided? Are all needed screws, cables, and connectors included? Take the time to check. Does the manual give instructions for multidrive systems? For putting the unit in unconventional locations? If not, you had better ask the salesman about these situations. Make sure he can arrange for support if you run into problems.

Switch settings. Before the drive is ready to run, you must check the switches within your computer and any settings on the disk controller card. Read the documentation and make sure everything is ready before you plug your computer back in.

Software. Some hard disks include complete MS-DOS shells to improve your hard disk management. Some include only software to format the disk. Some include nothing. Think about what's important for you.

Cost

Cost was saved for last because all other factors should be considered first. Decide what you need and want, then think about how you will pay for it.

Internal drives averaged about $22 per megabyte in 1986; hard disk cards ranged from $20 to $40; external drives were about $30. Interestingly, the cost per megabyte is fairly constant regardless of disk size. In other words, economies of scale don't apply to hard disks. Removable drives cost more than fixed drives, but even the cost of these units is dropping quickly. Speed always costs more.

Where to Buy

Three disk drive sources make sense:

- Buy from the local full-service vendor who originally sold you your system. Ideally, this firm knows your hardware, software, and your needs. They should be able to recommend an appropriate hard disk, install it, and answer your questions. You'll probably pay full price. If your vendor is good and you're not

Table 2-3. Hard Disk Card Size Requirements

Manufacturer	Drive	Slots	Mb	Power Connection	2nd*	List Price	Notes
Basic Time	Hardpack	1,2,3	20	Direct	Yes	$ 599	Includes 1Dir DOS shell; requires 1 short PC slot, 2 short XT slots.
CMS	Plus 21	1,2	21	Bus	No	$ 695	Requires only a single slot in slot 1 behind speaker.
Express Systems	3060	1, 1½	30	Direct	Yes	$ 795	
Express Systems	6060	2 or 3	60	Direct	No	$1095	
I2 Interface	I2Card30	2 for PC	32	Direct	Yes	$ 895	
Mountain Computer	DriveCard 30	1, 1½	30	Bus	Yes	$1195	
Plus Development	HardCard 20	1	21	Bus	No	$ 895	
Standard Brand	Flash Card-30	1½	32	Direct	Yes	$ 495	

* Second column indicates whether card supports an additional drive.

particularly interested in digging for answers yourself, it's worth it. Financial experts recommend buying all your insurance from one broker; this is the microcomputer equivalent.

- Buy from any local vendor who'll install and service the drive. It's worth paying extra if you don't want to install it yourself.
- Buy from the cheapest possible source, probably a reputable mail-order house in a state where you don't have to pay sales tax. Be prepared to handle your own installation or factor in the cost of hiring a consultant to do it for you.

Checklist

Below is a checklist to help you determine your hard disk needs. Simply fill in the blanks with your estimate of the quantity of information you have to deal with, and the bottom line will give you an approximation of the capacity required.

	Number of Disks	Characters per Disk	Total Characters
System files & utilities			
DOS	______	______	______
Operating environments	______	______	______
RAM-resident support programs	______	______	______
Compilers & developmental tools	______	______	______
Other utilities	______	______	______
Total:	______	______	______
Program files{1}			
Word Processing	______	______	______
Spreadsheets	______	______	______
Database/File Management	______	______	______
Accounting	______	______	______
Project Planning	______	______	______
Other: ______________	______	______	______
______________	______	______	______
______________	______	______	______
______________	______	______	______
Total:	______	______	______

Priority Data files{2}

Word Processing			
Spreadsheets			
Database/File Management			
Accounting			
Project Planning			
Other:			
Total:			

Productivity Software{3}

Word Processing			
Spreadsheets			
Database/File Management			
Accounting			
Project Planning			
Other:			
Total:			

Future Growth{4}

Word Processing			
Spreadsheets			
Database/File Management			
Accounting			
Project Planning			
Other:			
Total:			

Totals:

System files & utilities ____________
Program files ____________
Priority data files ____________
Future growth ____________

Total required space: ____________

Other data files ____________

Total desirable space: ____________

Disk Unit: ____________ Capacity: ____________
Type of Unit: ____________ Avg. Access Time: ____________
Supplier: ____________ Price: ____________
Comments: ____________

For each factor listed below, assign two values between one and five. One is lowest; five is highest. The first value indicates how important this factor is to you. The second value is this unit's rating on this characteristic. The score is the product of the two. Example: the Flying Disk 2000 rates a 5 on speed. Speed has level 3 importance for you. The score would look like this:

	Rating	**× Importance**	**= Score**
Speed	5	3	15

	Rating	**× Importance**	**= Score**
Absolutes {5}			
Compatibility with your current system	______	5	______
Compatibility with future plans{6}	______	5	______
Reliability	______	5	______
Total{7}	______	______	______
Resource demands			
Demands on the system's power supply	______	______	______
Internal space requirements	______	______	______
External (desktop) space requirements	______	______	______
Cost	______	______	______
Installation time	______	______	______
Total	______	______	______
Performance			
Speed	______	______	______
Storage capacity	______	______	______
Removability	______	______	______
Automatic park mode	______	______	______
Shock resistance	______	______	______
Documentation	______	______	______
Software	______	______	______
Total	______	______	______

Totals:

Absolutes:	______
Resource demands:	______
Performance:	______
Grand Total:	______

{1} Include only programs and files sent to you by the software vendor, not your own data files. Don't include tutorial or learning programs unless you're sure you want them on the hard disk.

{2} Data files are your own files. In this section, include only data files you're sure you want to keep on the hard disk. Count each file only once, even if it fits in more than one category.

{3} Count only the software you want to keep on the hard disk. If you know you don't want it, don't count it.

{4} Project two years ahead. Include all program files plus data files you know you'd like to keep on the hard disk. Of course you won't have accurate numbers. Estimate.

{5} As these can have no relative value (they are either true or false) they are each given an importance rating of five.

{6} Can the drive support a second drive? Can it be a second drive? Does it leave enough expansion slots for add-ons?

{7} If the total is greater than 50, transfer to Absolute score near bottom of page. If it is 50 or less, stop. Don't buy this unit.

Chapter 3

Installation and Care

Installation and Care

You've selected your system and your hard disk. You're eager to put them to use. What now? This chapter is about getting started: finding the right place for your system, installing your hard disk, preparing it for use, and keeping it running.

Find the Right Home for Your System

You probably thought your decision-making ended when you finally selected your hard disk. It didn't. Now you need to decide where to put this delicate electronic instrument.

Even a floppy disk–based computer needs care. However, a hard disk system is far more susceptible to harm. Spills, dust, or bumps can all damage the disk.

> Just as in real estate, location is important for computer systems.
>
> - Keep the system close enough to be convenient, yet out of the way when not in use.
> - Unless you bought a portable computer, don't plan on moving your system frequently.
> - Put the keyboard at optimal typing height, 25–26 inches.
> - Put the monitor at comfortable eye level. In particular, don't put it too high or your eyes and neck will strain.
> - Don't put your system near known danger from liquids, dust, direct sunlight or excessive heat.
> - Make sure the main system box is stable and out of the path of jiggles, bumps, and blows. Don't keep your printer on the same desktop.
> - Take precautions against static electricity, including but not limited to a good antistatic mat.
> - Use placement accessories and/or specially-designed computer furniture to make your system easy and conventient to use.

One common arrangement is to keep the computer on a typing stand or on the "return" portion of an *L*-shaped secretarial desk. Both these surfaces put the keyboard at the correct typing height. But neither is wide enough for both the computer and the keyboard, unless you use placement accessories. Even the end of a desk's return can be used effectively as a workstation.

An entire industry is ready to help fit all your computer gear comfortably into your workplace. The following accessories are examined below:

- Vertical system stands
- Keyboard drawers
- Monitor stands
- Monitor valets
- Line conditioners

Vertical System Stands

Your computer and hard disk are designed to work either vertically or horizontally. Since the system box is by far the biggest part of your computer's footprint, it makes sense to move it out of the way if space is limited. Also, moving the system box lowers the monitor to a better viewing height. Finally, if your printer sits on your desk, you'll want to move your system to insulate it from jiggles caused by the printer's head movement.

You can put the system on the floor near your computer work area, wedged against a desk or wall. However, it's much safer to use a specially-designed vertical stand. Whether or not you use a vertical system stand, insure that the fan and vents are unobstructed. Avoid having your system touch the desk, so it won't be jostled.

Computers range from four to more than seven inches high. If you're buying a stand, make sure you have the right aperture size (or buy one that adjusts). Stands range from simple U-shaped saddles to the deluxe full-system AT enclosure from IBM. Prices range from $30 to $150.

When you move your system onto the floor, you'll also need to extend your cables. Cable extenders can be purchased for keyboards, monitors, parallel printers, and serial devices such as mice or laser printers. They cost about $25 each.

Keyboard Drawers

Keyboard drawers are just high enough to hold your keyboard. They let you pull your keyboard out when needed and slide it out of the way when not in use. They effectively widen a narrow work surface and create additional kneespace.

Keyboard drawers come in two basic configurations:

- Surface-mount, for a narrow typing return or typing stand. Here they effectively widen the desktop so your computer will fit. Since these surfaces are already at correct typing height, this type of pullout drawer sits on the top surface. It can raise your monitor to a more comfortable viewing height.
- Hanging, for a full-height surface. Desks and tables are usually 29½ inches high, too high for comfortable typing. This style of keyboard drawer attaches to the bottom of your desktop. It lowers the keyboard to the proper typing height.

Monitor Stands

Premium monitors, such as the NEC MultiSync, provide tilt and swivel adjustments so you can view them from a comfortable angle. You can add tilt and swivel adjustment to any monitor with a monitor stand. These stands also raise the monitor a few inches to a better viewing height. Monitor stands cost as little as $25.

Portable computers, such as the Compaq Portable II, are fine for moving files and tools to various work sites. However, once you get there and set the portable system up, you'll discover that their screens are too low for comfortable viewing. Recognizing this, manufacturers now offer oversized tilt and swivel stands for about $100. These stands will accomodate the entire portable computer. A less flexible but more affordable solution to the portable computer dilemma is a printer stand turned backward. A printer stand will probably cost about $20.

Monitor Valets

A monitor valet functions much like a dentist's tool tray. Although it's fairly expensive (around $125), it frees a large amount of desk space. You can adjust it so your monitor is as close or far away as you desire. You can even swing your monitor completely out of the way when it's not in use.

The monitor valet requires that you route the monitor video and power cables through the valet's arm, so cable extenders are necessary. If you elect to place your keyboard on the pull-out rack provided, you'll need to lengthen its cable as well.

The valet swivels extensively but has no height or tilt adjustments. Adding an inexpensive monitor stand provides these adjustments.

Line Conditioners

An engineering firm purchased a microcomputer system for their office in an industrial park. The computer worked fine around the clock for months. It provided drafting and data entry support during office hours. It generated engineering analysis reports at night. Then it began to abort its overnight processing once or twice a week. Service calls and extensive diagnosis at a repair center uncovered no problems. Still, the system continued to fail at night.

Their computer consultant decided to stay in the office overnight until the reason was uncovered. Early in the morning of the second night he heard a thunderous hum, the lights flickered, and the system's monitor went blank. Rushing outside, he found that the noise came from a nearby building. Inquiries the next day revealed that the neighboring company had recently begun testing some rebuilt industrial turbines. They did the testing at night to avoid interfering with adjacent businesses. The engineers put in a line conditioner that same day, and the problem was solved.

The utility companies pump electrical power through a grid to serve many simultaneous users. As total demand varies moment by moment, so does the supply. Power companies switch additional transformers on and off as the overall demand varies. The constant change of supply and demand creates an uneven flow of electricity to your computer equipment.

Momentary surges of electrical current may greatly exceed the rated capacity of your computer. Too much current overloads transistors and diodes, shortening their lifespan and eventually destroying them. Surge protectors limit the amount of electricity that reaches any attached equipment. Dips in the electrical supply are called *brownouts*. Brownouts can create a reversed electrical flow and can damage circuits.

Electrical motors inject spurious signals (*noise*) into power lines. This noise can confuse integrated circuits, causing malfunctions within your computer. Power line filters prohibit the noise from entering any attached equipment.

Line conditioners provide both surge protection and power line filtering. Well-designed conditioners do more than prolong equipment life. They also make using the equipment more convenient. Some units sit beneath the monitor, raising it to a more comfortable viewing height. Line conditioners with power switches on the front panel let you power up without reaching behind your computer.

A line conditioner places the power control for your entire system in a single convenient panel. You may elect to power on your computer, monitor, and dot-matrix printer with a single switch. Using separate switches, you could turn on your modem or laser printer as needed.

Bad Vibes

Whether vertical or horizontal, make sure your system is out of harm's way. If it sits on the floor, insure that it is slightly recessed from adjacent furniture to avoid bumps. Keep it out of the path of kicks and coffee spills. It should not be next to the water cooler or under hanging plants. Give it adequate ventilation for cooling, and shade it from prolonged exposure to direct sunlight.

Don't put your system where you'll be opening drawers that will knock against it. Since typewriters and printers bang and jiggle a great deal, put them on a separate table, or put your system on the floor.

Some hard disks are more vulnerable than others. Nonremovable Winchester-type hard disks are sealed against smoke or dust damage, but may still be susceptible to bumps.

Static Electricity

Static electricity poses one of the greatest dangers to your system. A single spark of static electricity—the kind you feel in overheated rooms in winter—can easily reach 5,000 volts. Anything over 30 volts will probably destroy a computer chip. Static electricity is largely related to weather and therefore out of your control: Lower humidity equals high static electricity potential. However, you can recognize the problem and take some basic precautions.

First, get in the habit of touching something metallic before touching your system. A metal chair or desk is ideal, but even a nongrounded metal object will help. In extreme environments, you can use a static dissipation device—essentially a grounded metal plate.

Second, buy an antistatic mat. These contain special metallic threads which will ground you. The mat should be large enough to cover the entire area where your chair or your feet are likely to be.

Third, for extreme cases, try lowering the environment's static electricity potential. Synthetic materials and low humidity are the two most controllable factors. Consider replacing synthetic carpets with natural fibers. Avoid leather-soled shoes and synthetic-fiber clothing. And install a room humidifier. These complicated changes are no panacea: You'll still need to use an antistatic mat, and you'll have to cultivate the habit of grounding yourself before touching the computer.

To Move or Not to Move

Portable computers abound. Many portable computer workstations (specialized computer desks with casters) are available. Hard disk systems are better able to withstand movement than they were a few years ago. In particular, most drives now have self-parking heads which automatically lock into a safe position when you turn off the system. However, moving increases the probability that you will bump your system or shake something loose.

Lifting a corner of your computer system can slightly flex the motherboard, causing cards to shift within their slots to create short-circuits. If you don't move your system, you don't run the risk.

If you decide to make your system movable, you can minimize the danger by insuring that all internal hardware is fully secured. Keep your system on a sturdy cart or table with large casters. Check the floor and eliminate rough spots in the journey. If you will be rolling from carpet to smooth floor, for example, or from a high floor to a low one, install metal plates to ease the transition.

Install a power-down batch file and make it a habit to run it before you turn off the machine. This batch file should call a disk-parking routine such as those discussed later in this chapter.

Preparing to Install Your Hard Disk

If your local computer store has already installed your hard disk, you can skip this section. If you're a computer novice (or a hardware novice), you may be tempted to skip this section, to play it safe and hire someone to install your hard disk for you. That's all right if you don't mind paying the price, but you should know that installing a hard disk is easy.

You don't need anything fancier than a flat-bladed screwdriver and the ability to follow instructions. Pliers and/or nutdrivers are useful, but not necessary. The less hardware expertise you have, the more satisfying is the realization that you can install your own hard disk. Study the preinstallation steps below, then read the appropriate section for your type of hard disk drive.

Removing Your System's Cover

PCs were designed for easy service. Using a flat-bladed screwdriver or hex-nut driver, you can remove the computer system cover. This is a four-step process.

- Turn off your computer.
- Detach any cables leading into the rear of the system. This includes cables to the keyboard, monitor, and any external devices such as printers, mice, and so on. Don't unplug the power cord yet. You'll need it attached when you discharge static electricity later. Label each cable as you unplug it and label the connection it came from. Otherwise you may have trouble reassembling your system later.
- For ATs only: There may be a brown vanity cover on the back of the system. If so, separate its Velcro attachments. Also, turn the front key to the unlocked position.
- Remove the cover retaining screws from the rear of the system. Figures 3-1, 3-2, and 3-3 show their locations on PCs, XTs, and ATs, respectively. Unscrew them and put them in a safe place, such as a labeled envelope.
- Slide the cover off. Grasp it on either side of the system and firmly but gently pull it forward. On a PC or XT, it will stop sliding when the rear of the cover is about two inches from the front of the chassis. When this happens, tilt the cover up slightly and continue to pull, as in Figure 3-4.

Figure 3-1. IBM PC Cover Retaining Screws

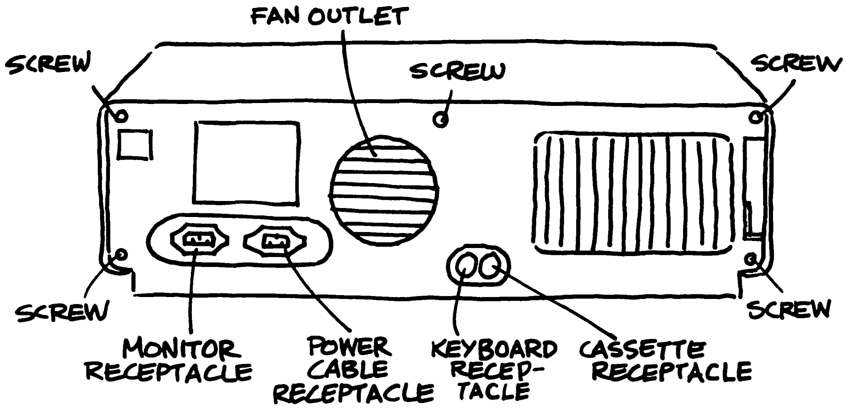


Figure 3-2. IBM PC XT Cover Retaining Screws

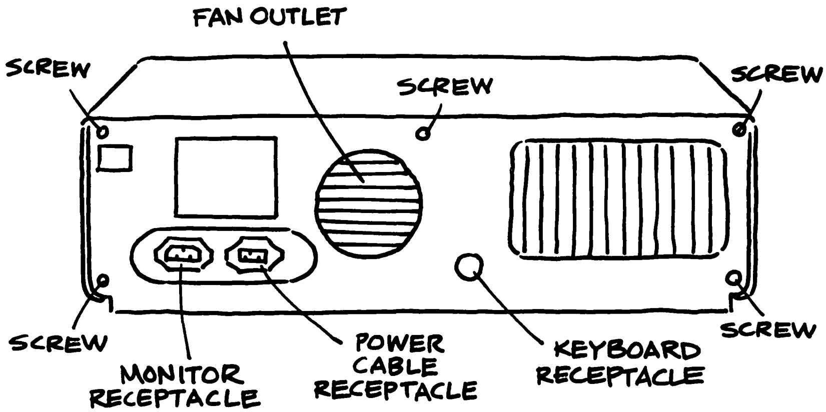


Figure 3-3. IBM PC AT Cover Retaining Screws

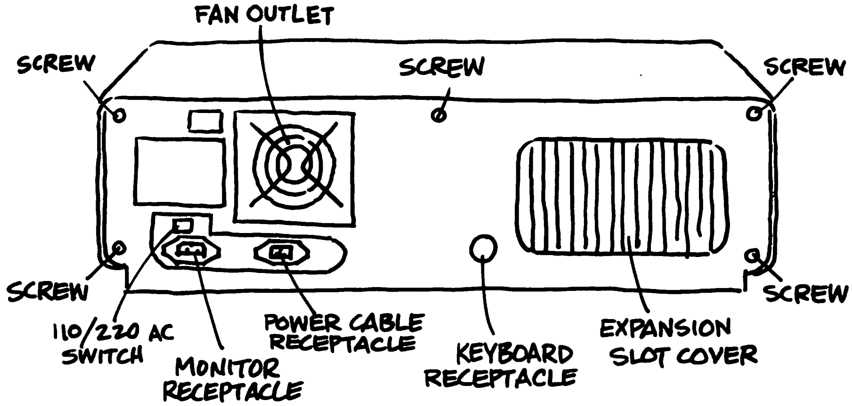
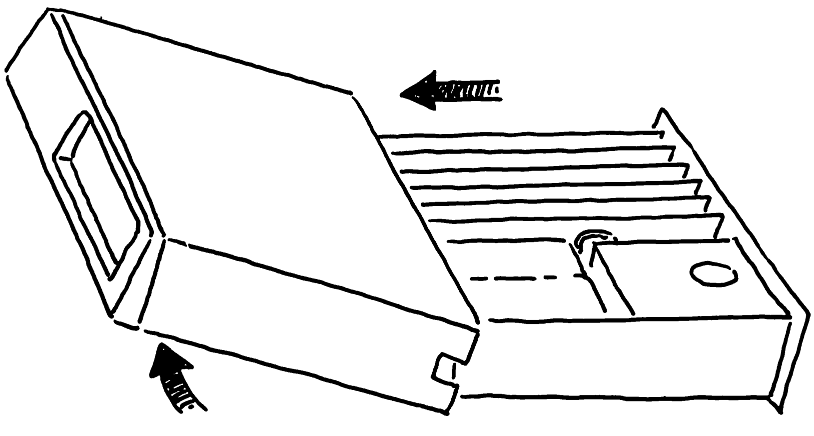


Figure 3-4. PC or XT Cover Removal



Once you have removed the cover, you can see the inside of your system. Look, but don't touch. You must eliminate static electricity before you can handle your system's internal parts.

Discharging Static Electricity

Before handling electronic equipment, you must always discharge any static electricity pent up in your body. This is extremely important. Even a small amount of static electricity is far too powerful for the integrated circuits within your computer. You can do hundreds of dollars of damage in far less than a second.

There are many ways to discharge static electricity safely. Here is one of the simplest procedures:

- Insure that the power cord is still attached to both the computer and the wall outlet. Verify that the computer is turned off.
- Touch the metal enclosure of the computer's power supply (refer to Figures 3-5 and 3-6). This sends the static electricity in your body to the ground where it can do no harm. The electricity moves from your body, through the power supply and power cord to the ground in the electrical outlet.
- While touching the power supply, touch any tools you will be using, as well as the hard disk drive you will be installing. This dissipates any static electricity within them.
- Finally, remove the power cord from the computer.

It is now safe for you to touch, move, and adjust the components within your system.

How to Install Expansion Cards

The PC, XT, AT, and compatible computers have an *open architecture.* These systems provide standardized connections that are linked to the computer's internal communication and control channels—its *bus.* By plugging expansion cards into these bus connectors, you can alter the capabilities of your computer. Figure 3-7 depicts the bus connectors within a PC AT. PCs and XTs are similar, but they only have a single connector per expansion slot.

Figure 3-5. Interior Layout of PC and XT

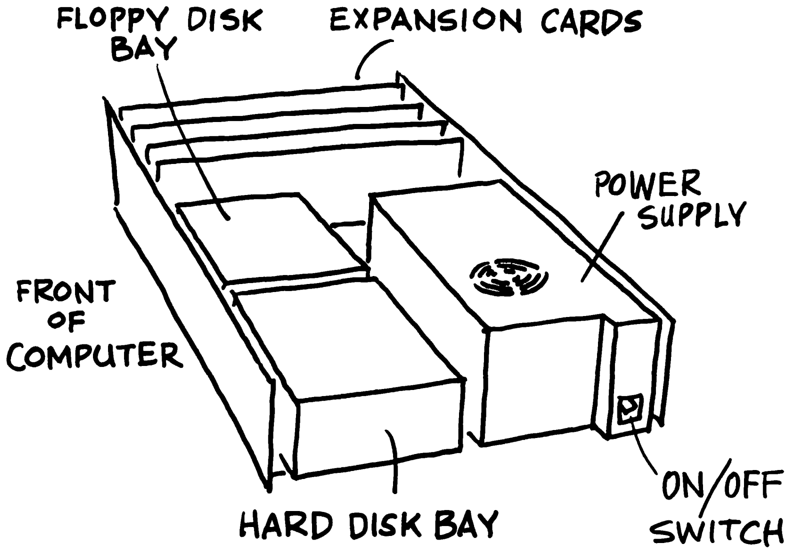


Figure 3-6. Interior Layout of PC AT

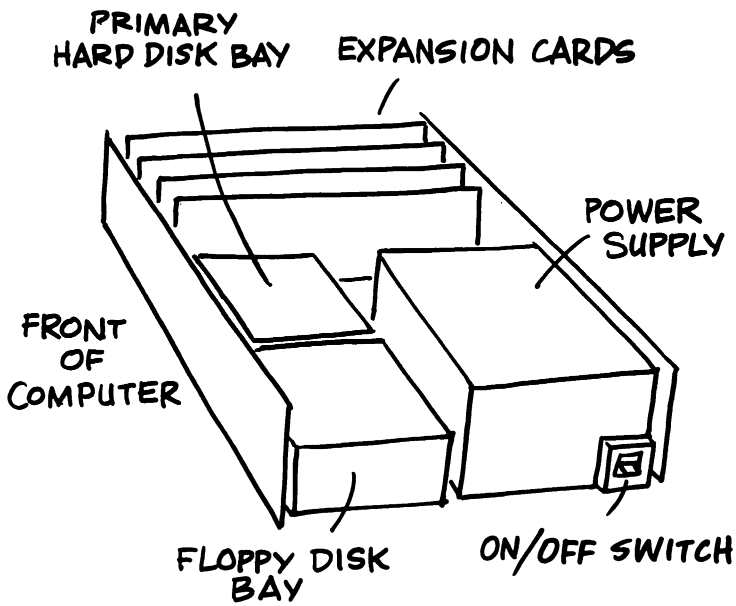
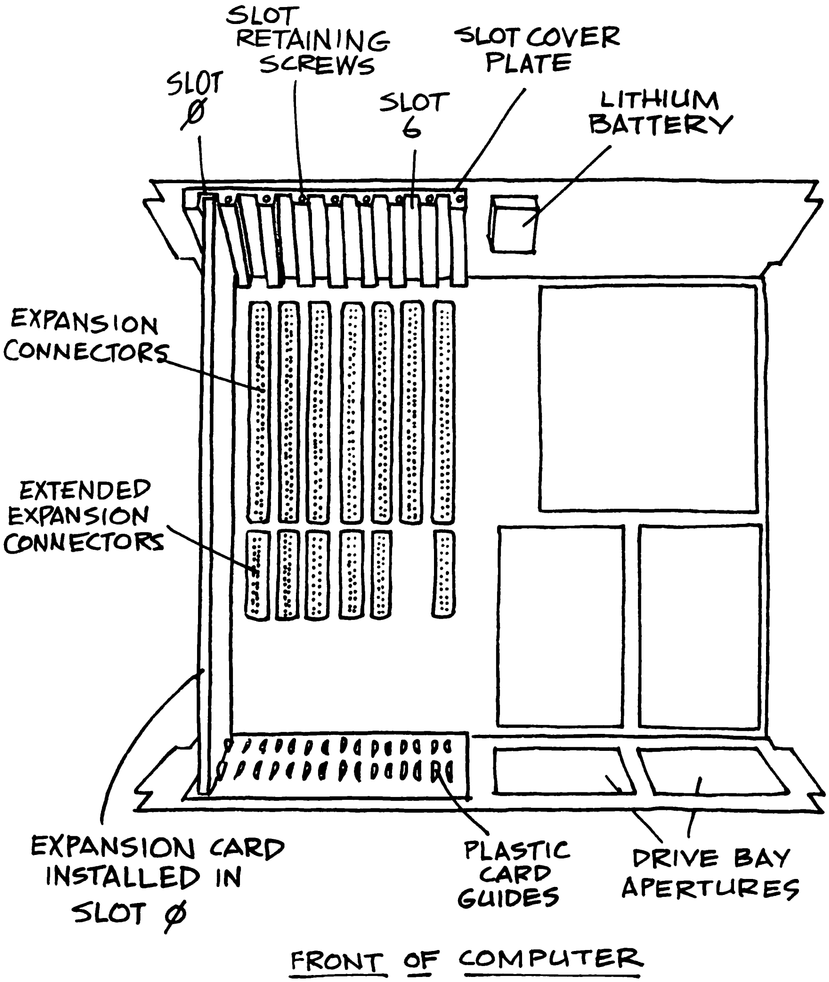


Figure 3-7. Bus Connectors Within a PC AT

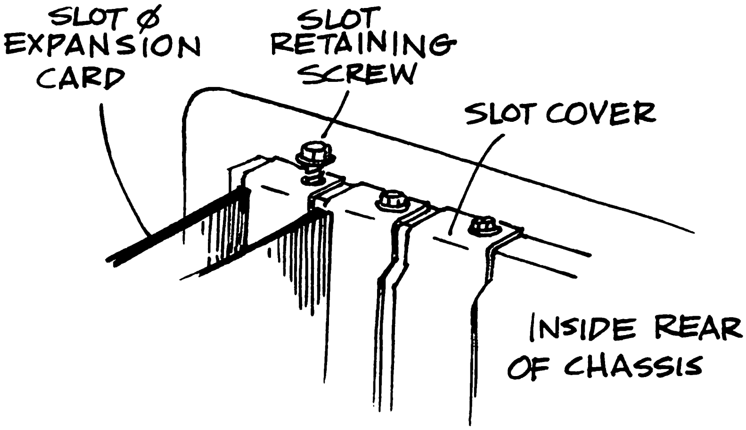
Expansion cards plug into slots on the PC AT motherboard.



Expansion cards have a lip that slides into the bus connectors. Some expansion cards are very short, extending from one-third to three-fourths the length of the computer. They are known as *half-cards.* Other cards extend from the inside of the front of the machine to the expansion slots at the rear; these are *full-size* cards. Such full-length cards are inserted within plastic card guides on the inside of the chassis front.

As depicted in Figure 3-8, expansion cards and slot covers are anchored at the rear of the machine. A slot retaining screw holds each card or slot cover in place. Turning the screw counterclockwise loosens and removes it. Extract and install these screws with care.

Figure 3-8. Slot Cover and Expansion Cards at Rear of the Computer



One of Murphy's laws states that bread always lands butter-side down. Similarly, a dropped slot-retaining screw will always disappear into your computer, where it can cause short-ciruit damage. Frequently, you must lift the computer upside-down and shake it to expel the errant screw.

The original PC used 50-pin bus connectors. The PC AT needed additional control lines, so IBM added an adjacent 34-pin extended connector to the bus. The 50-pin connector signals remained identical between the PC, XT, and AT.

Many cards designed for the PC and XT can be inserted into any slot within AT-style machines. A few, however, cannot. Even though IBM's Enhanced Graphics Card (EGA) has no 34-pin extended expansion connection, the bottom of the card extends below the top of the adjacent extended connector. Therefore, an EGA card can only fit within an AT's slot zero or slot six, where there are no extended expansion connectors (refer to Figure 3-7).

Most hard disk cards cannot fit into any of the PC AT slots. Their 3½-inch hard disk drives are typically thicker than a single AT slot. The drive's diameter is also large enough that, like the

EGA card, it extends below the top of the extended expansion connector. Therefore, while the hard disk card may itself fit into slot zero or slot six of a PC AT, the disk drive collides with the extended connector of the adjacent slot, preventing full insertion of the hard disk card.

Now that you know of static, slots, and screws, you are ready to install your hard disk.

Installing Your Hard Disk

This section will provide step-by-step instructions for installing hard disk cards, external drives, and internal disks. Obviously, every drive is different. When in doubt, use the specific instructions from your drive's manufacturer. However, most systems are similar enough for the following instructions to be useful.

Early versions of the IBM PC-1 had a Basic Input/Output System, Read Only Memory (BIOS ROM) that did not sense the presence of optional hardware controllers, such as the PC XT hard disk controller. As you can see in Figure 3-9, *Norton Utilities* SI (System Info) program displays the date of the BIOS ROMs.

Figure 3-9. System Information with *Norton*'s SI Program

```
C:\DIAGNOSE>si c:
SI-System Information, Version 4.00, (C) Copr 1984-87, Peter Norton

        Computer Name:  IBM/PC-AT
     Operating System:  DOS 3.20
   Built-in BIOS dated:  Friday, June 20, 1986
       Main Processor:  Intel 80286           Serial Ports:  2
         Co-Processor:  None                Parallel Ports:  1
 Video Display Adapter:  Monochrome
    Current Video Mode:  Text, 80 x 25 Monochrome
Available Disk Drives:  6, A: - F:

DOS reports 640 K-bytes of memory:
   288 K-bytes used by DOS and resident programs
   352 K-bytes available for application programs
A search for active memory finds:
   640 K-bytes main memory     (at hex 0000-A000)
    64 K-bytes display memory  (at hex B000-C000)

  Computing Index (CI), relative to IBM/XT: 9.0
       Disk Index (DI), relative to IBM/XT: 2.5

Performance Index (PI), relative to IBM/XT: 6.8
```

If your BIOS ROMs are dated earlier than October 27, 1982, they must be replaced before your computer will automatically run the operating system from your hard disk (*bootstrap load*). With the older ROMs, you must boot from a floppy disk and load a device driver (via the CONFIG.SYS file) before you can use your hard disk. Upgraded BIOS ROMs are available from any IBM authorized dealer.

Installing a Hard Disk Card

Hard disk cards are the easiest hard disk systems to install. Both the controller and the drive are mounted on a single card. The card fits into an expansion slot on the motherboard and draws its power through the slot. As a result, there are no power cables, no controller cables, and usually no switches to worry about. The only worry, in fact, is figuring out which expansion slot to use. Although the card sits in a single slot's connector, it may be two full slots wide. Or it may be one slot wide at the back but two slots wide in its front half. Only a few cards, such as the Plus Development HardCard, truly require a single slot.

Read the instructions for the hard disk card. Does it require one, one and a half, or two slots? A few cards come with recommended locations. For example, the Tandon BusinessCard 21 takes a single slot if you place it in slot zero. The Mountain DriveCard also takes a single slot if it is slot zero, but then you must relocate the PC speaker.

Once you know the card's requirements, study your PC. How many slots are available? If there simply isn't enough room no matter how you rearrange your expansion cards, you've bought the wrong hardware. Return it and start over. If you have enough room, but it isn't in adjacent slots, you'll need to move some boards around.

- For each board you will move, first identify the target slot.
- Unscrew the slot retaining screw from the target slot. Put it in a safe place outside the computer.
- Remove the slot-cover from the target slot.
- If needed, add a plastic guide for the target slot on the inside of the chassis front slot.
- Remove the retaining screw from the slot of the board you are moving. Put it in a safe place. Carefully grasp the top edge of the

board. Pull up with a gentle rocking motion to loosen the board from its slot.
- Still grasping the top edge of the board, firmly push it into its new slot. Make sure it's all the way in. Fasten the board securely by reattaching the slot-cover screw.
- Repeat these steps for each board that must be moved.

Once the target slot is empty, you are ready to connect the new hard drive card. If necessary, remove the slot-cover screw from the card's target slot as described above. Grasp the card on the top and insert it into the plastic card guide and the slot's bus connector. Firmly press its connector edge down into the bus connector; you may hear it snap into place. Install the slot retaining screw.

After the hard disk card has been inserted and screwed down, replace the system cover. Don't insert the cover screws yet—you may have to reopen the system and adjust something later. Then reconnect all cables and the power cord. Now you're ready to start formatting and partitioning. Skip ahead to the section on formatting and partitioning, below.

Installing an External Hard Disk

An external hard disk drive is slightly more difficult to install than a hard disk card, but simpler than an internal drive. The procedure consists of two basic steps:

- Install the controller card on the motherboard.
- Attach the external hard disk drive to the controller.

Set any configuration switches or jumpers on the controller board. The documentation will tell you if this is necessary. Consumer commodity items, such as mail-order hard disk kits, are generally preconfigured for standard options.

Switches have tiny plastic levers with two possible settings. Use a pencil, pen or screwdriver to change the settings. Usually an embedded *1* or *On* tells you which position closes the circuit. Jumpers are small insulated connectors that slide over two short adjacent posts. Disconnect jumpers with needle-nosed pliers or long fingernails, or pry them off with a small screwdriver. Use your finger tips to push them onto new pairs of posts, according to your needs as explained in your documentation.

Now insert the new disk controller card. Remove the slot retaining screw from the card's target slot. Grasp the card on the edges and insert it into the plastic card guide and the slot connector. Firmly press it into the slot, then reinstall the slot retaining screw.

Attach the external drive to the hard disk controller card by connecting the cable (or cables) to the external connector on the back of the computer. See your hard disk manual for details.

Replace the system cover, but don't insert the cover screws—you may need to reopen the system and adjust something later. Now reconnect all cables and the power cord. You're ready to start formatting and partitioning. You may skip to the section on formatting and partitioning, below.

Installing an Internal Hard Disk

An internal hard disk consists of a hard disk controller board, the hard disk drive itself, and two cables connecting them. PC AT hard disk drives do not normally require a disk controller card, but they do need two mounting rails. Nonstandard disk drives, such as those employing ESDI or RLL recording, do require controller boards.

Set any configuration switches or jumpers on the controller board. The documentation will tell you if this is necessary. Consumer commodity items, such as mail-order hard disk kits, are generally preconfigured for standard options. The PC AT controller board is already cabled and configured to support two hard disk drives.

Switches have tiny plastic levers with two possible settings. Use a pencil, pen, or screwdriver to change the settings. Usually an embedded *1* or *On* tells you which in which position the switch is closed.

You may be required to change jumpers in order to configure your system. Jumpers are small insulated connectors that slide over two short adjacent posts. Disconnect jumpers with needle-nosed pliers or long fingernails, or pry them off with a small screwdriver. Use the pliers or your fingertips to push them onto new pairs of posts.

Be sure that the cables are attached securely to the hard disk controller card. There are generally two ribbon cables: a wide data cable and a narrow controller cable. On each cable, a colored stripe

marks the pin one edge. This edge usually goes toward the bottom or rear of the controller card. Some cable connectors are keyed with a notch. Controller boards are often silk-screened with a legend indicating pin numbers. Check your hard disk manual for specific instructions.

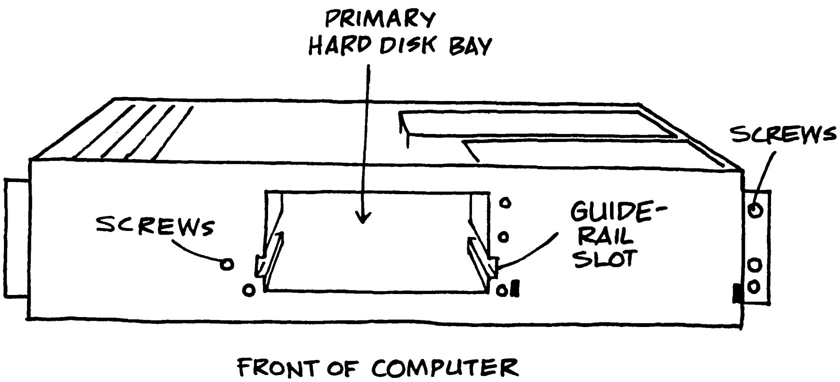
The hard disk controller cables should thread around the short floppy controller card (if any exists) and behind the floppy disk drive(s). For this reason, you will want to pick a slot that is close to the hard disk drive bay (refer to Figures 3-5 and 3-6).

Now insert the disk controller board. Remove the slot retaining screw from the card's target slot. Grasp the card on the edges and insert it into the slot's bus connector. If it is a full-length card, be sure that its forward edge is within the plastic card guide. Firmly press the card into the bus connector, then reinstall the slot retaining screw.

Move the hard disk cables into position at the rear of the hard disk bay. Avoid kinks and twists in the cables. Keep them flat. If you need to turn a cable at a right angle, fold it over at a 45-degree angle. If you need to shift it sideways a bit, make two angular folds.

As shown in Figure 3-10, the primary hard disk bay for the PC AT is in the center of the chassis front. The PC AT has room in its floppy disk drive bay for either three half-height devices or a single half-height and a single full-height device. This bay is located at the right of the machine. If you are installing a second hard disk drive into a PC AT, or if your hard disk uses removable media, you will put it in the lower portion of the floppy disk drive bay.

Figure 3-10. PC AT Primary Hard Disk Bay in the Center of the Chassis Front



A PC has no hard disk bay, so a hard disk would be installed into one of the floppy disk drive bays.

If the hard disk bay is empty, simply remove the faceplate hiding it from view. Typically this involves loosening two screws and lifting the faceplate from the chassis front.

If you are installing the first hard disk into a PC AT, you will find a metal box within the primary hard disk bay. This device is a load resistor. It connects to the power supply to give it something to do. Since your new hard disk will replace this do-nothing load, the load resistor can be removed and discarded. Disconnect it from the power supply. Remove the two screws securing the load resistor to the top of the hard disk bay. Extract the load resistor and set it aside.

If you must remove a floppy drive to make room for your hard disk, remove the second drive (the B drive). If you are installing a hard disk below your PC AT floppy drives, you must remove both floppies to gain access to the hard disk cables. To extract a floppy drive, follow these steps:

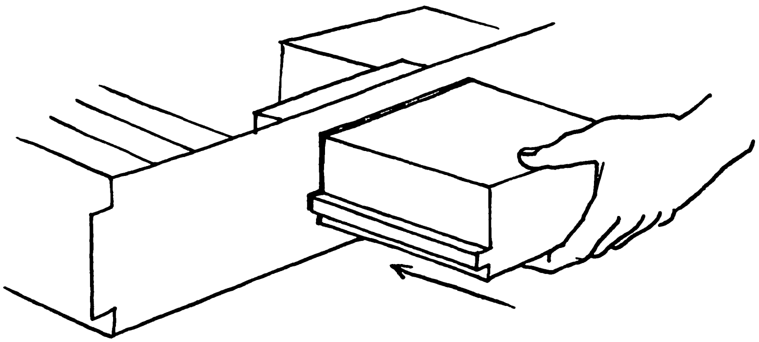
- Remove the screws holding the floppy drive in place. On a PC or PC XT, there will two or three screws attached to the bottom and sides of the drive. On a PC AT, there are two screws and clips beside the front of the floppy drive. Be especially careful not to drop screws into the system.
- Slide the drive forward far enough to reach the power supply wires and the floppy controller cable. Grasp each connector firmly, close to the drive. Wiggle the connectors to loosen them and gently pull them free. On a PC AT, also remove the black ground wire from the grounding lug.
- Slide the floppy disk drive out and set it aside. Remember which is the first drive, because it has a resistor network that is absent from the second.

For a PC AT, you may need to install *mounting rails*. These are plastic or metal bars that screw into the sides of the hard disk. To know what to expect, look at the rails on the floppy drive you just removed.

If no rails are installed and none came with your hard disk, stop. You *must* install PC AT drives using rails. Either take them from one of your floppy disk drives or get them from your hard disk supplier. Be warned that rails are hard to come by.

Now insert the hard disk into the drive bay, as in Figure 3-11. Slide the drive through the front of the bay, and push it in about two-thirds of its depth. Connect the data and controller cables from the hard disk controller card. These cables are keyed and will only fit one way. On IBM equipment, all connections are lettered, and the lettered sides go up. In any case, read the computer manual and the hard disk instructions carefully.

Figure 3-11. Inserting the Primary PC AT Hard Disk



Connect the cable from the power supply. It is keyed, and can only be connected one way. If there is a ground wire, as in the PC AT, slip it onto the flat-bladed grounding lug on the rear of the disk drive. Now finish sliding in the drive. Anchor it with the front or side screws. Make sure the cables are stable and out of the way, below the path of the cover when you slide it on later.

Add a faceplate if you are installing a half-height drive in a full-height slot. Faceplates usually snap right over the opening, or they may be attached by one or two screws.

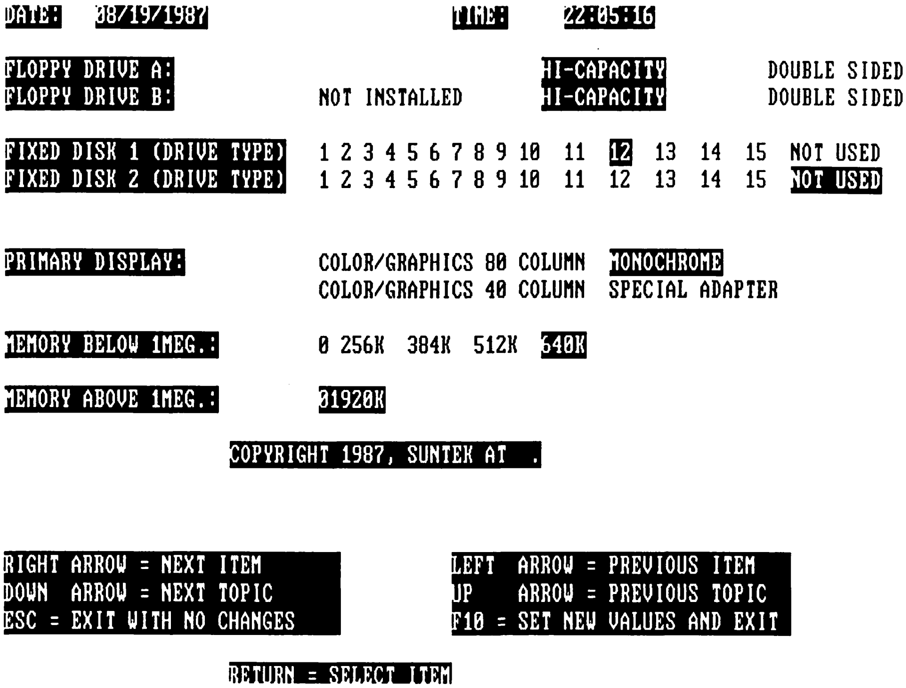
If you are replacing a PC or PC XT floppy drive with a hard disk, you must update the configuration switch settings on your motherboard. Check your computer's operating manual for the settings and switch location. Use a small screwdriver, pen, or pencil to correct the settings.

The PC AT stores its configuration settings in a battery-powered memory (CMOS RAM). This special memory is updated through software. IBM supplies the SETUP program for this purpose on their Diagnostics disk. You will have to reboot your system

to run it. Other vendors have written programs to set the configuration values. Figure 3-12 shows one that runs from within MS-DOS. Look on the front of the hard disk, at the manufacturer's label, to find the disk drive type. You will need to enter this information when you update the PC AT configuration later.

Figure 3-12. Program that Sets Configuration Values

PC AT configuration information is stored in battery-powered memory and is set by special programs.



After the hard disk card has been inserted and screwed down, replace the system cover. Do not insert the cover screws yet. You may have to reopen the system to adjust something later.

Reconnect all cables and the power cord. If you are installing your hard disk into a PC or PC XT, you're ready to start formatting and partitioning. If you're using a PC AT, you must first update your configuration information.

If you have installed the hard disk into a PC AT, turn on your

system and boot up from a floppy disk containing a SETUP program. Change the configuration information to indicate which drive type you have installed and whether it was the primary or secondary drive. Save the configuration. Now you can start formatting and partitioning.

Formatting and Partitioning Your Hard Disk

Once your hard disk is physically in place, you're ready for logical installation. This involves three steps:

- Physical formatting
- Partitioning
- Logical formatting

Physical formatting was probably completed before you received your disk. In addition, the vendor may already have partitioned and formatted the drive and may have even installed software for you. Make sure you know what has already been done. Both partitioning and logical formatting will destroy any files already on your disk.

If you're not sure what has been done, try booting from the hard disk.

- Open the door to floppy drive A.
- Turn on the system.
- If the system boots successfully, you will see the C: prompt. If so, logical installation is complete.
- If the computer prompts you to *Insert system disk*, logical installation still needs to be done.

Physical Formatting and Interleave

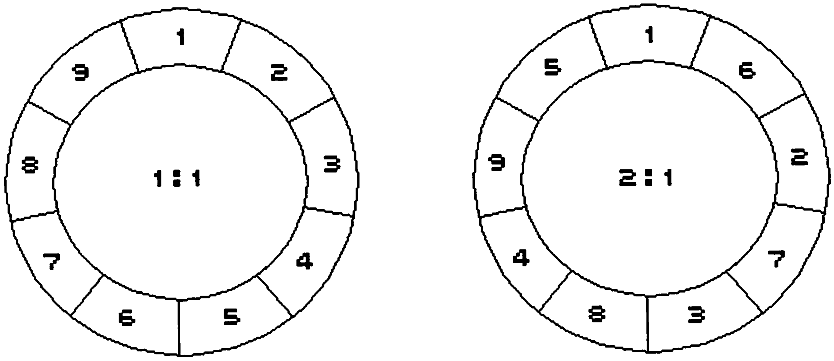
When you buy floppy disks, they are already physically formatted. Your hard disk probably was physically formatted, also. Most people never tamper with it. However, there is one reason you may want to: Inefficient formatting can slow your disk by 20 percent or more because of the *physical interleave* between sectors.

As you may remember, each track on your disk contains a fixed number of sectors. Theoretically, sequential sectors are physically next to each other. In fact, it's rarely true. Most computer systems aren't fast enough to read a sector, process it, and be ready

for the next sector before it spins under the read/write heads. To compensate, manufacturers mix up, or interleave, sectors. Refer to Figure 3-13 for two examples of interleaving.

Figure 3-13. Two Examples of Interleaving

Disk sectors can be physically sequential (1:1) or separated by interleaved sectors (2:1).



Theoretically, the drive manufacturer has already formatted for the optimal interleave. However, the right interleave depends upon the rotational speed of the drive spindle, the data transfer rate of the disk controller, and the computer's *clock* (processing speed). Many drives, even if purchased as part of a complete computer system, are therefore set incorrectly. As illustrated by the screen display in Figure 3-14, sector interleave has a dramatic effect on disk read/write speeds.

Changing the sector interleave may sound forbidding so early in your hard disk history. However, there's one big advantage to setting the right interleave now: Your disk is empty. If you wait and change the interleave later, after you've been using your hard disk awhile, backing up and restoring all your directories, programs, and data files will be a major project.

Finding and setting the right interleave once required advanced technical skill. Now it's easy—a few commercial disk utilities include interleave testing and reformatting. *HOPTIMIZE* from Kolod Research and the *Disk Engineer* from Prime Solutions are two examples. Both will tell you the best interleave for your system and, as a separate step, reformat as appropriate.

Figure 3-14. Effect of Sector Interleave on Disk Read/Write Speeds

Kolod Research's HOPTIMIZE *measures the performance of common hard disk sector interleaves.*

```
Formatting specified area at interleave    6 ...
Now read testing...
At interleave    6 the elapsed time for reading was    5 seconds

Formatting specified area at interleave    7 ...
Now read testing...
At interleave    7 the elapsed time for reading was    5 seconds

Summary is as follows...
```

Interleave	Total access time (rounded to nearest sec.)	Total clock ticks (normally 18.2 / sec)
1	13	241
2	2	31
3	2	44
4	3	58
5	4	71
6	5	84
7	5	95

Western Digital disk controller boards let you reset the interleave and reformat the drive using the DEBUG program that comes with MS-DOS. Once you know what interleave is best for you (generally 2:1 or 3:1), invoke the program within the disk controller's ROM with the following commands:

A:>DEBUG
-G C800:0005

Respond to the request for interleave with 2 for 2:1, 3 for 3:1, and so on. Then direct the controller to perform its low-level format. You are performing the same process as the disk drive manufacturer, using the same tools, optimizing for your particular computer.

Partitioning

After your hard disk is physically formatted, but before you format it logically and start using it, you must divide it into areas called partitions. You can have up to four partitions on a disk, or even more if you fool MS-DOS. Partitions serve two main purposes:

- They provide for different operating systems. PCs have an open architecture. This means, among other things, they were designed to work with a number of operating systems, including MS-DOS, Xenix, CP/M-86, and PC-IX. IBM gives each its own partition. You specify which operating system is in charge when the system starts up. This will be covered later.
- They work with large-capacity disks. As discussed in Chapter 1, your disk is divided logically into 512-byte sectors. Up through version 3.2, MS-DOS handles no more than about 64,000 sectors, or 32Mb, per partition. Therefore, if your hard disk is larger than 32Mb, multiple partitions will be necessary. Fortunately, it's possible for one disk to have several partitions for a single operating system.

In most cases, your disk will contain just one partition and one operating system—MS-DOS. Still, it's nice to know you have a choice. The simple case of one partition/one operating system will be discussed first, followed by a look at fancier options.

Creating the MS-DOS Partition

Even if your disk is 32Mb or smaller and MS-DOS is the only operating system you want, you still need to create an MS-DOS partition to satisfy system startup requirements.

When you turn on your hard disk system, a series of built-in instructions from the Read Only Memory, Basic Input/Output System (ROM BIOS) takes control. First the system checks and initializes its hardware. Then it starts looking for a bootstrap loader program. The bootstrap loader is a small program (512 bytes or less) that loads the rest of the operating system and starts the system functioning.

The startup program looks on the first surface, track, and sector of floppy drive A. If a boot disk is in drive A, it finds the bootstrap loader there. If drive A is empty or doesn't contain a boot disk,

the startup program next looks on the hard disk, if there is one.

Startup from the hard disk is slightly different. What the startup program reads from the first sector of cylinder zero, side zero, head zero, is not a bootstrap loader. Instead, it is a partition table and a program to read and use its contents. This program is often called the *partition loader* or the *master bootstrap loader.*

The partition table holds one 16-byte entry for each partition on the disk. The table tells the starting location, the ending location, and the operating system of each partition. It also identifies one partition as the *active partition.* The active partition is the one whose bootstrap loader will function at startup time.

The partition loader loads the bootstrap loader from the first sector of the active partition. The bootstrap loader loads the rest of the operating system, and your computer is off and running.

Now you understand why you must create at least one partition. Without it, there would be no entries in the partition table, and you could never boot from your hard disk. Indeed, you couldn't use your hard disk at all, since any disk space used must be part of a valid partition.

With this background complete, this is how to create a partition: You'll need the MS-DOS FDISK program, found on the System Master disk. As depicted in Figure 3-15, when the DOS partition is defined, it will exclude cylinder zero. Why waste an entire cylinder? The partition information is at the beginning of cylinder zero. If a program runs amuck and begins to destroy all the hard disk area it can find, you won't want the partition table to be within reach.

To define your hard disk MS-DOS partition:

- Boot MS-DOS and run the FDISK program.
- Create a new DOS partition by selecting choice one. Since this choice is the default, just press the Enter key.
- If the drive has already been partitioned, you will receive the message *DOS partition already exists.* If this happens, select the *Change DOS Partition* option and define the DOS partition to exclude cylinder zero and include whatever contiguous range of cylinders is available.
- If you are asked whether to use the entire hard disk for DOS, answer *N* for no. You will then be able to define exactly where your DOS partition will go.

Figure 3-15. DOS Partition Defined

Define the DOS partition to exclude cylinder zero.

```
Create DOS Partition



Partition Status  Type   Start  End Size
   1       N      DOS        1  613  613




Total disk space is 614 cylinders.
Maximum available space is 614
cylinders at 0.

Enter partition size............:  613
Enter starting cylinder number..:    1

DOS partition created
```

- As you can see in Figure 3-15, the FDISK program tells you how many cylinders are available on your hard disk. Enter the partition size to be one less than the total disk space available, and enter the starting cylinder as one.
- Select the *Change Active Partition* option. As in Figure 3-16, select the DOS partition. This will cause the partition loader to select the MS-DOS bootstrap loader whenever the system starts up.
- Save your changes and exit from the FDISK menu. The hard disk light will glow briefly, then the system will restart and tell you to insert the DOS disk into drive A.
- After MS-DOS loads from the floppy disk, you are ready to format the hard disk.

The MS-DOS FDISK program will only create a DOS partition. To create partitions for other operating systems, you must use their partitioning software. If you have more than one operating system on disk, you must pick one operating system to boot from. To do so, use FDISK and pick option two, *Change Active Partition*.

If you want to use more than 32Mb of your disk for MS-DOS, you must create multiple DOS partitions. If your FDISK program is part of an MS-DOS version older than 3.3, you won't be able to

Figure 3-16. DOS as the Active Boot Partition

```
Change Active Partition



Partition Status  Type   Start  End Size
    1       N     DOS       1  613  613




Total disk space is 614 cylinders.



Enter the number of the partition you
Want to make active...............:[1]
```

format a disk larger than 32Mb. You will have to find software that will. Most large disks come with their own software. Also, commercial hard disk formatting packages such as *Vfeature Deluxe* and *SpeedStor* are available. Most of these packages will format disks larger than 32Mb.

These programs use two methods for accessing large disks:

- Increasing the sector size.
- Fooling DOS into seeing the physical drive as several separate drives.

Changing Sector Size

MS-DOS programs have no problem working with a different sector size. This is because MS-DOS performs all the disk activity for them.

Normally a DOS sector is 512 bytes. Many large-disk programs change the size to anything from 1,024 (1K) to 16,384 (16K) bytes. Sixteen thousand (16K) sectors allow up to 1000Mb (known as a gigabyte or 1G) in a single DOS partition. Large sectors are inefficient, however, since every file occupies at least one cluster, which is made up of several sectors.

When you change the sector size of a drive, you must reformat the entire partition. This is a logical format, operating-system specific, and not the low-level hardware format where you define the disk interleave. Back up all files and programs you have within the

partition being reformatted, because formatting will destroy all the information within the partition. If you have other hard disk partitions, they will not be affected.

Making Many Logical Drives on a Single Physical Drive

Version 3.2 and earlier MS-DOS only allows one DOS partition per disk. Many programs bypass this limit by fooling MS-DOS into seeing a single physical drive as many logical drives.

Device drivers, such as ANSI.SYS, are the tools of this deception. Device drivers enable MS-DOS to use any nonstandard device. For example, MS-DOS only recognizes specific disk configurations. If the speed, size, or arrangement of platters and cylinders on your disk doesn't match one of the known patterns, you must use a device driver to give MS-DOS the needed facts.

Each device driver has a one-line entry in the CONFIG.SYS file on the root directory. (Don't worry if these terms are new. You'll learn about them in Chapter 4.) As part of MS-DOS initialization, the bootstrap loader reads CONFIG.SYS and loads all device drivers it references.

Device drivers for large disks contain code enabling MS-DOS to operate as if it were using several disks rather than a single large one. For example, MS-DOS may use a single physical drive C, as if it were separate drives C, D, and E. Since large disks are often nonstandard, the device driver also provides necessary configuration data.

Device drivers can fool MS-DOS in other ways. *VFeature Deluxe* includes a SPAN option, the opposite of breaking a single disk into several apparent disks. Spanning fools MS-DOS into seeing two physical drives as one large single volume. Again, the device driver provides the means of deception.

Figure 3-17 shows a CONFIG.SYS containing multiple device drivers, only one of which is for hard disks. The device drivers are: Quarterdeck's *QEMM-386* for defining AT-style extended memory as paged EMS memory in 80386-based computers, Golden Bow's *Vfeature Deluxe* for defining a 60Mb drive as three 20Mb drives, MicroSolutions' *UNIFORM* for accessing various floppy disk formats, and Mouse Systems' *PC MOUSE* interface.

Figure 3-17. CONFIG.SYS with Multiple Device Drivers

```
C:\>type config.sys
device=c:\hdwr\qemm.sys frame=c400 dma=128 context=16
device=c:\hdwr\fixt_drv.sys
device=c:\hdwr\uniform.sys
device=c:\hdwr\msmouse.sys /2
files=60
buffers=17
break=on
shell=c:\command.com /e:992 /p

C:\>
```

Formatting Your Hard Disk

Logical formatting is similar to formatting a floppy disk, and nearly as easy.

Boot from the MS-DOS System Master floppy disk in drive A. Type the command

FORMAT C: /S

MS-DOS will prepare drive C to receive MS-DOS files, and then it move the operating system files onto the drive. Assuming you're using MS-DOS 3.0 or later—if you aren't, you should be—you'll see the message *WARNING, ALL DATA ON NONREMOVABLE DISK DRIVE C: WILL BE LOST!*
Proceed with Format [Y/N]?

Earlier versions of MS-DOS did not include this warning—they simply reformatted the entire disk, destroying any data on it. Answer by pressing *Y* and then Enter. You will see the message *FORMATTING . . .*

Formatting will take a few minutes and may take up to half an hour. When it completes, you'll see two messages: *Format complete* and *System transferred.*

At this point, MS-DOS will display the total number of bytes available on your disk. Now verify that your disk drive is all right. Open the drive A door. Press Ctrl, Alt, and Del at the same time. The system should boot. You'll know it worked when you see the date prompt. Type in the date, press Enter. Type in the time, and press Enter again. You should see the *C:* prompt. If so, you have been successful. If not, close the drive A door, reboot, and format again.

Now verify that all the cards you moved while installing your hard disk still work. If you have two video monitors, use both of them. Print with your parallel printer. Send some data to your serial laser printer or modem. Move your mouse. Once you are satisfied that all cards work, go ahead and replace your system's cover screws.

Checking Out Your New Disk

Theoretically, you're ready to go now. However, practical experience dictates the next step: check out the drive. Electronic and hardware problems are most likely when equipment is new. Before investing precious time loading up your new drive, make sure it works. Two check-out procedures are recommended:

- Keep your system running continuously for a week or so after installing your drive. Since hard disks spin continuously, you'll be giving it nonstop exercise. This may expose any electronic or mechanical problems early, while the drive is empty and under warranty.
- Run a good sector-checking program. MS-DOS's FORMAT program finds and marks the most obvious bad sectors. However, you don't want to use marginal sectors and then lose data later as a result. A good sector-checking program will thoroughly exercise the disk and catch potential problems. *The Disk Technician, Norton Utilities, HTEST* and *Mace Utilities* all include good diagnostic/repair programs. A good diagnostic program takes time, but it's worth it. *The Disk Technician's Monthly Test,* which is what you'd run on your new drive, takes nearly five hours to test a 32Mb drive on an XT. Just plan ahead and run it when convenient.

Figure 3-18 shows *Norton*'s DT (DiskTest) program finding the same bad sectors that the MS-DOS FORMAT had already marked. DT lists all files that are using bad sectors. You would then copy the affected files to another area, delete the originals that use bad sectors, rerun the diagnostic program to mark the now-unused bad sectors, and then attempt to repair your files (see Chapter 10).

Figure 3-18. Searching for Bad Sectors

Norton*'s DiskTest detects bad sectors and prevents them from being used for data.*

```
C:\NORTON>dt
DT-Disk Test, Version 4.00, (C) Copr 1984-87, Peter Norton

Select DISK test, FILE test, or BOTH
Press D, F, or B ... D

During the scan of the disk, you may press
BREAK (Control-C) to interrupt Disk Test

Test reading the entire disk C:, system area and data area
  The system area consists of boot, FAT, and directory
    No errors reading system area

  The data area consists of clusters numbered 2 - 10,393
    5,668th cluster read error: already marked as bad; no danger
    5,669th cluster read error: already marked as bad; no danger
    5,670th cluster read error: already marked as bad; no danger
    5,671st cluster read error: already marked as bad; no danger
    5,672nd cluster read error: already marked as bad; no danger
    5,916
```

Keeping Your Drive Healthy

Now that you have survived installation and break-in, and now that you are up and running, how do you stay that way? If you've picked the right location to set up your computer or external hard disk drive, you've already taken the most important step. Isolating your drive from bumps, jiggles, external magnetic fields, temperature extremes, dirt, liquids, smoke, and other pollutants will extend its life considerably. An antistatic mat protects your system from static electricity. A dust cover, faithfully used, gives obvious protection.

In general, you want to keep the area around your system as clean as possible. At the very least, don't let things pile up so much that you cut off the cooling air flow through the vents. Sometimes excessive cleanliness can cause problems, though. Vacuum cleaner motors can produce strong, destructive magnetic fields. Whole data libraries have been damaged by a single overzealous cleaner. Be careful.

Turning Your System On and Off

Don't turn your system on and off more than necessary. Like other electrical equipment, both your computer and your hard disk receive extra stress during startup. To minimize stress, minimize startups. Turn your system on and off only once a day. Your entire system uses only about as much power as a couple of 100-watt lightbulbs, so it won't cost much. In fact, it'll probably save you money through lower repair bills.

Leaving an image on your monitor for hours can burn it into the screen's phosphor coating. To avoid damage, reduce the screen brightness whenever you leave your system. A better, automatic solution is a memory-resident screen blanking program.

SCRNSAVE, a public domain program, clears the screen after two minutes without a keyboard entry. When you're ready to start work again or want to see what's happening, pressing any key restores the screen. *SCRNSAVE* has one big drawback: It doesn't understand graphics modes. If it dims your screen while you're running the graphics database *Reflex*, it freezes your system. This is very frustrating if you've spent the last hour adding records to a memory-resident database. A commercial product, *Cruise Control*, handles the graphics mode correctly.

Leaving your machine on all day exposes your hard disk to more potential damage from electrical spikes, surges, sags, brownouts, or failures. These fluctuations can cause head crashes, damaging whatever is beneath the heads. You need a program that moves the heads safely away from the data area whenever practical.

The Disk Technician's *SafePark* is a memory-resident program that does this. It uses less than 700 bytes. The first time you run it, it creates a special safety zone. Whenever your hard disk is inactive for more than seven seconds, it moves the heads to the safety zone. Thus, even if the heads crash, no damage results.

Backup Power Supplies

If electrical problems are common in your area, you may want additional protection. After all, with so much of your data on a hard disk, you have more at stake when power surges or fails.

Several levels of power regulators are available. Surge and noise protectors are the simplest. They range from $20 to $100.

Some are nearly useless, so shop carefully. Good ones cut out most noise and block power surges and spikes, typically up to 6000 volts. Some also protect against static electricity. Most have four to six outlets and a single on-off switch. Hence another minor advantage of these and all power supplies is the ability to turn your entire system on or off with a single switch.

Surge protectors guard against too much power, but not against too little. The next step up is a standby power system (SPS), which features a battery backup system. Power failures trigger the battery backup, giving you time to power down your system without damage or data loss. Standby power systems cost $500 and up, and vary in quality as much as in price. All SPSs include a connected series of sealed, maintenance-free lead-acid batteries (similar to the one in your car) plus a battery charger. A good SPS should also include:

- Low noise and small size
- Surge and spike protection
- Fast response to line fluctuations
- Power sag responsiveness
- Adequate power
- Sine-like wave patterns

Let's look at these in more detail. A good SPS is small and quiet. Until recently, SPS boxes were bulky and awkward because of their large battery packs. However, some units that provide the same protection are now small enough to fit beneath your monitor. For example, the DataSaver 400 from Cuesta Systems is only two inches high.

SPSs need cooling. Pay attention to the noise level of the cooling fan. As in other areas, there is a wide variation in fan noise. Some fans are quiet. Some are so unpleasant that you wouldn't want them near your desk.

Some SPSs let surges and spikes through unchanged. This is dangerous and completely unacceptable for anyone willing to spend money protecting against power loss.

Fast response insures the SPS reacts before your computer does. PCs shut down automatically when they detect a power fluctuation of 20 percent or more. Most PCs take at least 20 milliseconds to respond. Some IBM PC XTs, though, react to power

fluctuations much faster. These hyperreactive machines use a Zenith power supply manufactured in Mexico.

Power sags or brownouts should trigger SPS backup power. After all, a 20 percent power change will shut down your PC. Unfortunately, many SPSs respond only to blackouts and recognize significant brownouts too slowly or not at all.

Adequate backup power is a must. The SPS power rating for your computer alone should be at least 50 percent greater than the rating on your power supply. A power supply that delivers 135 watts to your computer will require 200 watts of power from the outlet. The power supply itself uses a certain amount of energy, and some energy is lost due to inefficiencies inherent in transformers. Your SPS will have to provide all the power needed to run your computer, or it will be useless. Also, when considering the purchase of an SPS, you must allow adequate power for your monitor, printer, and any other devices that might be running when the lights go out.

High-energy items such as laser printers shouldn't go on the SPS at all. You can fortify most systems with additional, external batteries if their rated power isn't enough. Make sure the unit's battery charger will charge the external batteries as well its own.

The power level also determines how long the battery backup will last. Most systems last about 15 minutes, enough time to save your work and power-down normally.

Batteries produce direct current, so all SPS's include a DC-to-AC invertor. Cheap invertors produce square waves that can overheat your system. Regular AC current arrives in sine waves. You don't need a perfect sine wave from your SPS, but you need a close approximation. Try to get a look at an oscilloscope tracing of the power output of any SPS under serious consideration.

The fullest power protection comes from an uninterruptible power supply (UPS). A UPS provides continuous battery power rather than just standby power when needed. It recharges the batteries continuously when current is available. It operates like an SPS during power sags or failures. Its two main advantages are:

- There is never a gap in power because the mode of power delivery never changes.
- The UPS insulates completely against spikes and surges because all line current feeds into the batteries rather than your computer.

Most SPS manufacturers call their units *UPSs*, so shop carefully. Also, you should ask the same questions about size, noise, power ratings, wave forms, and so on when shopping for a UPS as when shopping for an SPS. UPSs tend to cost more than SPSs—starting around $900. Some UPS units use the more compact under-the-monitor design. These units cost around $1,000.

Moving Drives

Moving your system presents great potential hazard to your delicate hard disk. If something the size of a smoke particle can damage your disk, think what could happen if you picked up your system and banged it around.

Because the disk is so fragile, all disk manufacturers provide a way to park the heads. *Parking* means moving the heads to a safe area, away from all data, and insuring that they stay there until the system is turned on again.

On early personal computers, you physically parked the heads by screwing them into place. Nowadays, many hard disks park automatically whenever you turn off your machine. If yours doesn't, or if you're not sure, run a parking program before moving the machine. You should have received one with your system.

On most IBM-compatible equipment, the program is called *SHIPDISK.EXE*. On IBM systems it's on the diagnostic disk, one of the set of operating system disks. You can run it from floppy disks, or you can copy it to your hard disk and run it there. When you do, you'll see the hard disk's red light come on; then the program will tell you to turn off your system.

Turning on the system frees the drives. This is handy, unless you turn it on before the move, then forget to run the parking program again.

Whenever there's danger of bumping during a move, repack the system in its original packing materials. You should keep the cardboard guards for floppy drives and use them whenever the system is moved.

Maintenance

Now that you've installed your own hard disk, you may feel ready to take on the unit's maintenance as well. The best advice is to leave hard disk maintenance to a professional. Hard disk maintenance requires proper tools, diagnostic aids, schematics, environment, and experience. Your job is to provide a good operating environment, to maintain organized data (Chapter 4) and regular backups (Chapter 8), and to learn the warning signs of disk problems (Chapter 10). Leave the repairs to experts.

Conclusion

The aim of this chapter has been to get you and your hard disk off to a good start. There can be great satisfaction in installing a hard disk yourself, preparing it, optimizing it, and setting up an appropriate, safe workplace for your hard disk system. To further simplify your setup, three checklists are included at the end of this chapter. These checklists concern:

- Creating a safe home for your system
- Moving your system safely and easily
- Selecting the right power regulator for your system

In the next chapter, you'll start organizing your disk. You'll learn about directories and trees, and you'll create some directories of your own. You'll create a CONFIG.SYS file on your root directory. You'll see elementary batch files, and you'll create some of your own. By tackling the entire question of disk organization in a structured, systematic way, you'll save yourself hours of frustration later on.

System ______________________ Date ________________________

Your Name ____________________

Is This System's Location:	*Yes*	*No*
Close to where it's needed	______	______
Out of the way when not needed	______	______
Stable: free from jiggles	______	______
Protected by an antistatic mat	______	______
Out of the path of		
kicks	______	______
opening desk drawers	______	______
other bumps	______	______
Away from		
cigarette smoke	______	______
excessive dust	______	______
other airborne pollutants	______	______
liquids	______	______
excessive cold	______	______
excessive heat	______	______
direct sunlight	______	______
excessive static electricity	______	______
Efficient in its use of space	______	______
Is the keyboard at proper typing height?	______	______
Is the monitor comfortably placed?	______	______

Hard Disk Drive

Manufacturer and model ________________________________

Speed and capacity ________________________________

Interleave, if known ________________________________

Where documentation is kept ________________________________

Installation date ________________________________

Vendor name and phone number ________________________________

Other useful information ________________________________

Moving	*Yes*	*No*
Do we move this system as little as possible?	______	______
What is required to park the hard disk heads?		
nothing: self-parking	______	______
parking software	______	______
physical manipulation	______	______
If parking software is used,		
Program name	______	______
Location	______	______

Any special instructions for parking:

____

____

____

____

Where packing materials are kept for this system ______________________

____

A Power Regulator Checklist (Optional)

Name of protective device ______________________________

Type (surge protector, Standby Power Supply, true Uninterruptible Power Supply)

____

	Yes	*No*
Does this unit protect against	______	______
surges	______	______
spikes	______	______
noise	______	______
brownouts	______	______
power failure	______	______
Is protection adequate as far as		
wattage	______	______
response speed	______	______
noise level	______	______
size	______	______
approximation to sine waves	______	______
Can this unit recharge external as well as internal batteries?	______	______

Chapter 4

Organization

Organization

This chapter is about planning.

In the last chapter, you physically installed your hard disk and completed the formatting necessary to use it. Those steps were somewhat generic, varying according to hardware and not your individual needs. In this chapter, you'll start organizing your hard disk system to reflect your personal work-style.

Some basic concepts and common sense tips will be introduced. However, your specific disk organization will depend on your needs, your data, and your environment. This chapter takes you beyond the cookbook stage. The principles of good cuisine can be summed up in a few words, but it's up to you to do the cooking.

To make good planning decisions, you need information. This chapter focuses on four topics you need to understand:

- Directories
- Environments, PATH, and related commands
- Batch files
- AUTOEXEC.BAT and CONFIG.SYS files

You're probably eager to start loading and using your disk. However, you will benefit from reading this chapter thoroughly before loading any software. Otherwise, you may find yourself reorganizing what you've already done.

What are Directories?

When you first started school, all your important papers fit easily into a small stack on your desk. As you grew older and took on greater responsibilities, your stack of papers grew. Today you may have many cubic feet of papers to organize.

If your level of organization has kept pace with quantity in your paperwork, you probably use filing cabinets and files rather

than keeping all your papers in a pile. And, if your collection of paper is very large, you probably have some system for cataloging where the important papers are, such as warranties and receipts you will need for tax purposes. Even if you don't have such a system, you are probably aware that things would be simpler if you did.

Disk drives are like filing cabinets. Some have a single drawer, while others have many. Directories are the disk equivalent of manila folders in those drawers. Directories let you organize your programs, support tools, and files for maximum safety, simplicity, and ease of use.

Boost performance with the organizational features of MS-DOS.

- Decide on an overall structure.
- Keep the root as empty as possible.
- Have separate directories for MS-DOS (\DOS), hardware support (\HDWR), system utilities (\UTIL), and batch files (\BATCH).
- Use short directory names and consistent filenames.
- Make your directories now, and put the most used items first. Copy your .BAT, .EXE and .COM files before loading data files.
- Use PATH, APPEND, JOIN, SUBST, and ASSIGN to broaden accessibility.
- Use batch files to tailor directory access for faster program loading, shorter file searches, and a simpler PATH directive. Batch files make it easy to use full pathnames, and full pathnames cut down directory searches.
- Use CONFIG.SYS to boost performance. MS-DOS lets you fine-tune your system with disk buffering and break-checking.

Directory Organization

Every disk automatically contains one directory, called the *root directory.* The MS-DOS FORMAT command creates the root directory as part of its logical formatting. The directory contains one entry for each file stored in it. A directory entry is 32 bytes long and contains:

- Filename/extension
- File size, in bytes
- Date and time when the file was created
- Starting disk location for this file
- File attributes—a directory is itself a file, but with a special Directory attribute. Other attributes are Read-Only, Archived, Hidden, and System.

Except the file attributes, each of these items is listed on the screen when you give a directory command (*DIR*). Figure 4-1 shows a sample directory listing. To see the directory contents for the directory you are pointing at, the current directory, simply type **DIR.**

Figure 4-1. Sample Directory Listing

```
C:\>d:

D:\>cd sn

D:\SN>dir

 Volume in drive D has no label
 Directory of  D:\SN

.            <DIR>      1-31-87   3:50p
..           <DIR>      1-31-87   3:50p
README           6400  11-25-86   1:28a
DEMO         <DIR>      1-31-87   5:41p
DEMO     BAT       28  11-06-86   7:49a
SNOFF    COM      112   8-03-86  12:22p
SN       EXE    59238   9-25-86  11:23a
SNPRINT  EXE    10900  11-17-86  12:18a
SN       HLP     4096  10-10-86  11:27a
EXAMPLE  TXT      896   1-02-86  12:13p
       10 File(s)    903168 bytes free
```

Notice that the directory listing does not include detailed location data for each file. This information resides in the file allocation table, or FAT, discussed in Chapter 1.

The FAT gives the status of every sector on a disk. The FAT entry of each sector points to another sector within the FAT. This is called a *linked list.* The FAT contains a linked list of bad sectors that were marked as unusable (by FORMAT or some other program). It also has a list of free sectors—those available for use by files. Finally, the FAT contains a linked list for each file on the disk.

A directory entry points to the file's first sector entry within the FAT. From there, the file's entire location is available by following its linked list. When MS-DOS erases a file, it marks the directory entry as empty. MS-DOS then adds the now-erased file's linked list of sectors to the available list within the FAT. This is faster than physically erasing the data from the sectors. It also allows you to unerase files, as will be discussed in Chapter 10.

Tree-Structured Directories

Think of a directory as a manila file folder. Figure 4-2 shows that Digital Research's Graphics Environment Manager, GEM, even uses pictures of manila folders as *icons,* or symbols, for disk directories. A directory listing is like an index sheet in that folder. The folder contains a number of documents. The index sheet shows which documents are in the folder and when they were put there.

This is useful, but not useful enough. You also need a way of organizing all the manila folders. In MS-DOS, tree-structured directories are the key to large-scale organization.

As mentioned above, every disk automatically has one directory—the root directory. The root directory is specified as a reverse-slash with or without a preceding drive identifier (\ or C: \ ). The root can contain files. It can also contain other directories, called *subdirectories.* If you think of the root directory as the trunk of a tree, then its subdirectories are branches.

Furthermore, subdirectories can have subdirectories of their own. This is the concept of nesting: Directories can nest inside other directories and can in turn have their own nested subdirectories. There is no limit, besides physical space, to the number of directories or the levels of nesting. Figure 4-3 illustrates the tree structure of a hard disk directory.

Figure 4-2. GEM Representation of Disk Directories

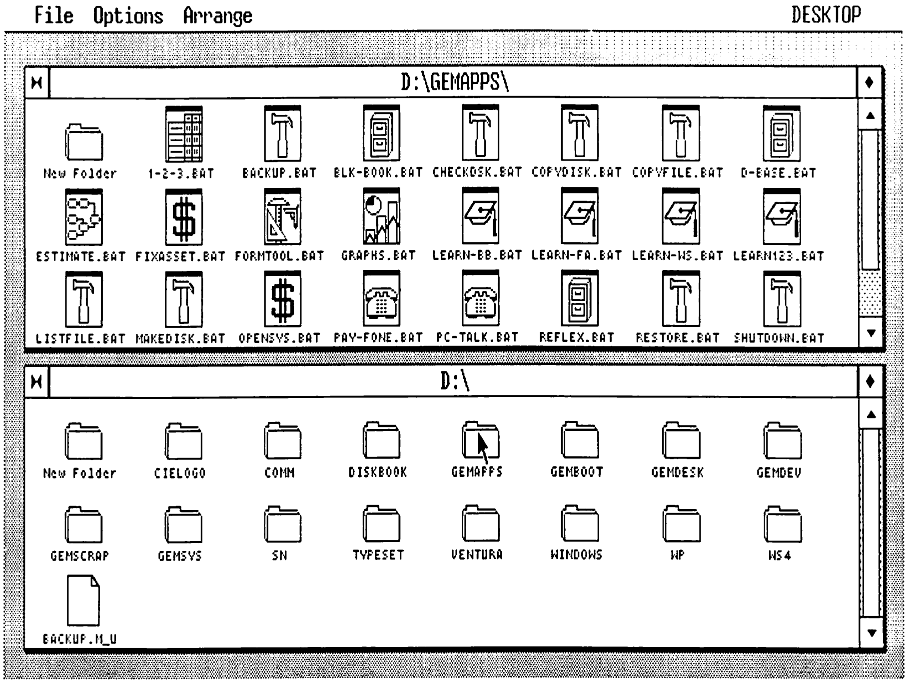


Figure 4-3. Directory Structure

```
D:\>vtree
┬─ CIELOGO
├─ COMM ──────────┬─ KERMIT
│                 ├─ MIRROR
│                 ├─ MOVE-IT
│                 ├─ PROCOMM
│                 └─ XTALK
├─ DISKBOOK
├─ GEMAPPS
├─ GEMBOOT
├─ GEMDESK
├─ GEMDEV ────────┬─ DOC
│                 ├─ SOURCE
│                 └─ TOOLS
├─ GEMSCRAP
├─ GEMSYS
├─ SN ───────────── DEMO
├─ TYPESET
├─ VENTURA
├─ WINDOWS ──────── PIF
├─ WP ───────────── DATA
└─ WS4 ───────────┬ LETTERS
                  └ TUTORIAL
```

The public-domain VTREE (Visual TREE) *program graphically depicts the directory structure.*

Notice that *subdirectory* is a relative term. It refers to a directory nested within another directory. Every directory except the root is a subdirectory. Whether you call it a directory or a subdirectory depends on the context. If its nested status is important, call it a subdirectory. Otherwise, it's simply a directory.

Parent Directories

Look again at Figure 4-1. Notice the entries with *<DIR>* next to the names. As you might have guessed, these are directories.

The first two directory entries have dots as filenames. Dotnames are shorthand. A single dot (.) indicates the current directory; a double-dot (. .) is the *parent directory.* In Figure 4-1, . means *D:* \ *SN* and . . is the *D:* \ directory. The third directory entry, DEMO, is a subdirectory of the current directory. Therefore, *D:* \ *SN* \ *DEMO* and . \ *DEMO* are synonymous.

You can also use . . in identifying directories. Imagine your hard disk has the structure depicted in Figure 4-3. Imagine that you are logged into \WS4 \TUTORIAL. LETTERS is another subdirectory of \WS4 and . . \LETTERS is a shorthand way to refer to \WS4 \LETTERS.

We can classify directories by their distance from the root. Subdirectories of the root are called level one directories. Subdirectories of level one directories are level two directories, and so on. The root is the parent of level one directories, which are parents of level two directories.

As you can see from the first column in Figure 4-3, COMM, DISKBOOK, and SN are level one directories. They are *siblings* to each other, and *children* of the root directory. In the second column, KERMIT and DEMO are level two directories. KERMIT is a child of COMM, and SN is the parent of DEMO. The DISKBOOK directory has no children.

Names and Paths

Directory names follow the same rules as filenames:

- The name can be up to eight characters.
- The extension can be up to three characters.
- The name can include letters, numbers, and certain special characters: ! @ # $ % ^ & ( ) - _ { } ' '

- You cannot have two files or directories with the same name in one directory. (You can have identical filenames in different directories.)

Because there can be many levels and branches of directories, there are strict rules for identifying directories:

- You can give a drive identifier. If you don't, the current drive is assumed.
- The root directory is \ (backslash).
- \ separates all levels of directory names.
- If a directory name doesn't start with \, the current directory is assumed to be the parent.

A directory's full chain of ancestors is called its *path*. (This is not the same as the MS-DOS PATH command, which will be discussed later in this chapter.) For the following examples, refer to Figures 4-1 and 4-3. D:\ is the root directory of drive D. \DISKBOOK is a child of the root on the current drive. DEMO is a child of the current directory, whatever it may be. D:\SN\DEMO is a child of SN, which is a child of the root on drive D.

Directory Commands

MS-DOS provides three commands for working with directories. *MKDIR* creates a new directory, *RMDIR* removes a directory, and *CHDIR* moves you to a different directory. All three are internal commands. This means they are accessible anytime you see the MS-DOS prompt. All three identify directories using the rules discussed above.

Creating directories. MKDIR (MaKe DIRectory) creates a new directory. You can also type *MD* rather than MKDIR. If the current directory is to be the parent of the new directory, you simply enter

MD ***directoryname***

MS-DOS then creates a new subdirectory in the current directory.

If you want to make a directory with a different path, type the complete path along with the directory name. The path, except for the about-to-be-created directory, must already exist. For example, if you already have a level 1 directory named LOTUS, to make a new subdirectory of LOTUS, named BUDGET, enter

MD \LOTUS\BUDGET

If you are in \WP\DATA\LETTERS and want to make \WP\DATA\MEMOS, type

MD ..\MEMOS

On the other hand, if you want MEMOS to be a subdirectory of LETTERS, simply type

MD MEMOS

Moving to another directory. CHange DIRectory, typed as CHDIR or CD, moves you to another directory. It also changes the current directory (also known as the *logged directory*). It uses the rules already discussed for specifying pathnames.

If you're not sure what directory you're in, typing the CHDIR or CD command without an argument will show the current directory of the current drive.

Each drive has its own current directory. You can see the current directory for another drive by giving just the drive identifier, as in Figure 4-4, which also illustrates using the MS-DOS prompt to constantly display the current directory.

Figure 4-4. CHDIR and PROMPT Used to Determine Current Directory

```
C>d:

D>cd
D:\DISKBOOK

D>cd c:
C:\BATCH

D>prompt $p$g

D:\DISKBOOK>cd ..

D:\>c:

C:\BATCH>
```

Unless you change it, the MS-DOS command prompt is simply the currently logged drive followed by a greater-than sign. The logged drive is the one that is used if no drive is explicitly specified. Referring to the first line of Figure 4-4, when you're logged onto drive C, the unadorned prompt is C:>.

The command entered on the first line logs onto drive D, so the prompt changes to *D:>*.

Once you start using subdirectories, it's important to know where you are. Files which seem to disappear mysteriously can often be found in either the root directory or up one directory level from where they belong. This is not a bug in the system, but instead is most likely the result of confusion over which directory you are in.

Seeing a constant reminder of your current directory can eliminate some confusion. The MS-DOS PROMPT command lets you change the form and content of the MS-DOS command prompt. It provides numerous options to generally rearrange the MS-DOS prompt format. For now, what's important is that you can change the prompt to include the current drive and directory. To do so, follow the example in Figure 4-4 and type

PROMPT pg

The dollar sign ($) separates PROMPT parameters. The *p* option displays the current drive and directory; the *g* option displays the greater-than sign (>). This is such a useful option that many MS-DOS systems run it as part of their startup procedure, AUTOEXEC.BAT. For more details on the PROMPT command, check your MS-DOS Reference Guide.

Removing obsolete directories. The RMDIR or *RD* command ReMoves DIRectories. As a safeguard, MS-DOS only lets you remove empty directories. There are five steps to removing a directory (here called the *target directory*):

- Empty all subdirectories of the target directory.
- Remove all subdirectories of the target directory with the RD command.
- Use the DEL command to erase all files in the target directory.
- Move to a higher level directory or to another branch with the CHDIR or CD command.
- Use RMDIR or RD to remove the target directory.

A public-domain program, *KILLDIR* by Amy J. Goebel, provides the convenience (and risk) of allowing you to erase nonempty children directories. It will not work if the directory is already

empty, however. Figure 4-5 lists a batch file that combines RMDIR and KILLDIR to work with both empty and full directories. Appendix B fully explains the techniques employed within this batch job stream.

Figure 4-5. A Batch File that Removes Subdirectories

```
C:\BATCH>type eradir.bat
echo off
if "%1"=="" goto :NODIR
ctty nul
rd %1
killdir %1
ctty con
dir %1
goto :END
:NODIR
dir | find "<" | sort | more
echo+
echo You may only remove one of these subdirectories
:END
exit
```

Directory Strategies

Creating a useful directory tree-structure requires careful study and planning. There are a few basic and nearly universal rules. So before the discussion of difficult choices, here are the basics:

- Keep the root as empty as possible.
- Use short directory names and consistent filenames.
- Establish your overall structure.
- Make your directories now, and put the most used items first.
- Make separate directories for MS-DOS (\DOS), hardware support(\HDWR), system utilities (\UTIL), and batch files (\BATCH).

Keep the Root Clean

The root should contain only the bare necessities. MS-DOS starts every search for a file in the root directory. Extra clutter will reduce the hard disk performance you paid for.

The root should contain only its required files.

- AUTOEXEC.BAT, your startup batch file. Ideally, this consists of a single line that invokes STARTUP.BAT within your \BATCH directory.
- CONFIG.SYS, your system configuration file. Keep a backup copy of this in your \BATCH directory.
- The hidden MS-DOS files, MIO.SYS and MSDOS.SYS (or their PC-DOS equivalents, IBMBIO.COM and IBMDOS.COM). These files have a *hidden* attribute and therefore won't appear in directory listings. They are put here by the SYS command, but only if you first ran FORMAT with the */S* option to reserve space for them.
- COMMAND.COM, the MS-DOS command processor. You can put it elsewhere and reference it with the COMSPEC command. You can substitute your own version of COMMAND.COM via the SHELL command. But leave everything standard, because many programs are written to look for COMMAND.COM in the root of drive C.
- Your level one subdirectories.

Certain software vendors insist upon cluttering your root. Even Paul Mace, a respected MS-DOS guru, has his *Mace Utilities* maintain an image of the disk directory, BACKUP.M_U, in the root. Countless installation programs also write batch files into the root. These batch files can and should be moved out of your root, preferably to your \BATCH directory.

The \DOS, \HDWR, \UTIL and \BATCH Directories

Every bootable hard disk should contain four standard level one directories and a short search path.

- \DOS contains the operating system files, just as they come out of the box.
- \HDWR contains hardware-specific programs, such as device drivers and hardware configuration or test programs.
- \BATCH contains batch files and related *script* (redirection) files.
- \UTIL contains system support programs, such as pop-up tools, directory managers, and print spoolers.
- The corresponding search path is:

PATH . .;C:\BATCH;C:\HDWR;C:\UTIL;C:\DOS

The \DOS directory. The \DOS directory contains all MS-DOS external commands. (Some installations name this directory \BIN, in the UNIX operating system tradition.) Rather than copying MS-DOS files onto your root directory, as the MS-DOS manual advises, copy the MS-DOS disks here. The advantages of using a \DOS directory are:

- An uncluttered (and faster) root directory.
- Easy upgrade to a new version of MS-DOS. You merely transfer the new MS-DOS system to your hard disk with the SYS command, then you erase everything in the \DOS directory and fill it from the new disks.

> Use a \DOS directory to simplify upgrading MS-DOS.
>
> - Boot from the new System Master disk.
> - Put the new operating system onto the system area of the hard disk with the SYS command.
> - Log onto the hard disk.
> - Clear the old system from the \DOS directory with the ERA command.
> - Move the new MS-DOS files into the \DOS directory with the COPY command.
> - Remove the BASIC source programs, if you don't use them.
> - Make any needed changes to your AUTOEXEC.BAT and CONFIG.SYS. This will only occasionally be necessary.
> - Reboot from your hard disk.

The \HDWR directory. The \HDWR directory contains all those programs that are specific to the hardware within your system. Basically, any software that comes with a hardware component is a candidate for this directory. Device drivers, such as a mouse driver, or diagnostic software for an expansion board or external device would be appropriate for this subdirectory. Installation programs, however, should not be kept on your hard disk. Keep them on a floppy in your archives in order to conserve space. By keeping your hardware-specific support software in one place, you can easily remove unneeded programs when you change or remove hardware components.

The \BATCH directory. The \BATCH directory contains all of your system-wide batch files. Rather than include your word processing and spreadsheet program directories in your search path,

have short batch files in the \BATCH subdirectory that load these programs from their respective directories (see Appendix C for examples). \BATCH also holds any batch support files, such as the menu system screen displays from Appendix C. We recommend that you keep a backup copy of your CONFIG.SYS file here as well.

The \UTIL directory. The \UTIL directory (or \BIN \USR for UNIX systems) is the repository for most of the work-enhancing programs discussed in Chapter 5. Here is where you would keep system support programs, including your print spooler, your backup and recovery software, and your desktop accessories. How do you determine whether a program belongs in the \UTIL subdirectory? If the program works with more than one application package, it belongs.

Use Consistent Names

Names are a powerful organizational tool. Setting and following a consistent policy for all directory and file names will save you hours of hunting.

Use extensions on all file names. Some extensions are traditional or required:

- .TXT for ASCII text files
- .DOC for word processor document files
- .BAS for BASIC program files

Many programs automatically add extensions to filenames. For example, Ashton-Tate's dBASE adds the extension .BAK to original database files to differentiate them from new versions. Where you have a choice, add your own consistent, meaningful extensions.

Don't use extensions on directory names. If you always use extensions for files, this will immediately distinguish directories from files. Furthermore, omitting extensions will keep directory names shorter.

Keep directory names as short as possible, without making them meaningless. You'll be typing directory names often, and shorter names are easier. Also, MS-DOS limits the total number of characters in a complete file specification (drive, path, and filename) to 64 characters. You'll rarely even come close, but why take a chance?

Make directory names meaningful. Name your directories \LOTUS, \DBASE, and \WORD rather than \D1, \D2, and \D3, even though the latter are shorter. Often, software vendors have installation procedures that assume a given directory name. If so, use their name. That way you will avoid problems installing add-on software that is not flexible enough to use your directory name.

Be systematic with filenames. Make every character count. Keep all filenames for the same type of file the same length. For example, monthly expense files for Factory One might be FAC0101.EXP through FAC0112.EXP. This makes it easier to use a *template* to select a group of similar files.

MS-DOS allows *wildcard* characters within filenames. Many MS-DOS programs understand these ambiguous filenames are a request to process all files matching the template. For MS-DOS, the question mark (?) matches any single character, including a space. The asterisk (*) is equivalent to as many question marks as would fill out the filename or extension. For instance, to copy all the monthly expense files for Factory One to the floppy disk in drive A, you could type

COPY FAC01*.EXP A: /V

Likewise, to select January expenses for all factories, you would enter

COPY FAC??01.EXP A: /V

Consistent filenames simplfy wildcard references. They also help you to remember what a given file contains.

Establish Your Overall Structure

Directory decisions ultimately boil down to two questions:

- What is your fundamental organizing principle?
- How complex a tree structure do you want?

Your fundamental organizing principle. People typically organize their disk in one of two ways. The most common is a software-based grouping. Project grouping is an alternate structure. One of these two methods will probably suit your personal workstyle.

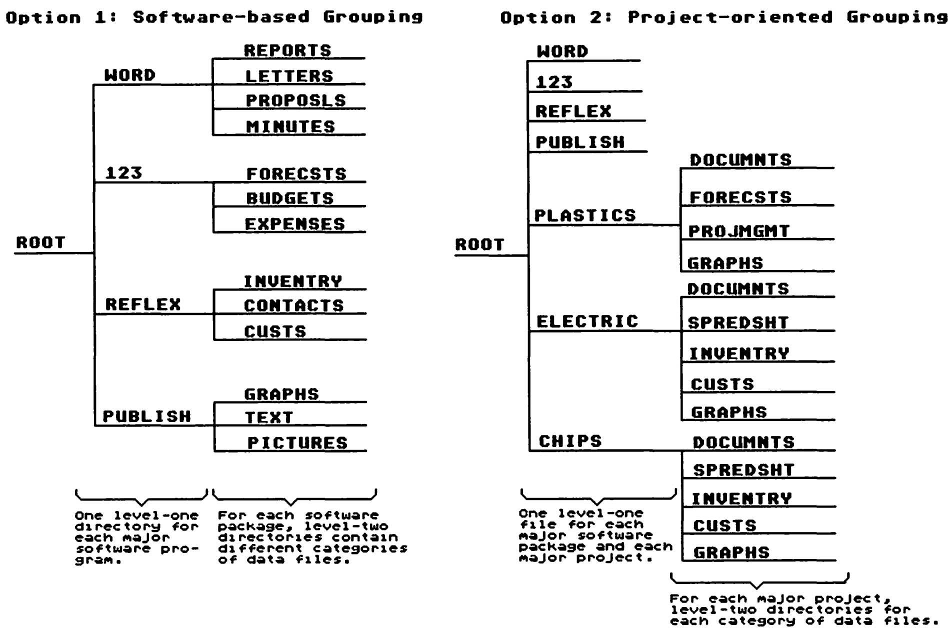
Software-based. Most commercial software packages presume that you have a software-based hard disk organization. In such a

scheme, you have a level one directory for each major software package. You put your data files for the program either in its directory or in its subdirectories.

Project grouping. If your company is set up along multiple product lines, you may want a different structure. This approach also works well for consulting firms with a few concurrent major projects. If it is a computer supply company, you might have three broad product lines: cabinetry plastics (PLASTIC), electrical power supplies (ELECTRIC), and integrated circuits (CHIPS). Each product area would have its own project plans, databases, spreadsheets, advertising copy, and memos. You'll frequently need to combine diverse files to produce finished documents. Rather than put CHIPS' databases in one directory and its spreadsheets in another, you may elect to put all of CHIPS' files together in a single directory. You would still have a single directory for each major software package, containing just the program and its support files.

Figure 4-6 summarizes both options. Choose project grouping only if you'll frequently combine information from diverse file types. Otherwise, a software-based structure is simpler to live with.

Figure 4-6. Two Ways to Organize Your Hard Disk



Person-centered directories are a project-oriented variation. If only one person uses each related set of files, consider making a separate directory or directories for each person.

You can also combine approaches. Within the word-processing or spreadsheet directory, you might have separate subdirectories for each person using the program (\WORD \SUE and \WORD \JOE), or for different types of files (\LOTUS \BUDGET and \LOTUS \QUOTES).

Complexity of structure. Some people insist on a simple structure. They want minimal directory nesting, even if it means putting program and data files in the same directory. They create separate directories for each software or project function. However, they don't add additional subdirectories unless the number of data files in a directory becomes completely unwieldy.

Others believe a slightly more complex structure is ultimately easier to use. If a set of files is logically distinct, they put that set in a separate subdirectory. This approach keeps directories small. If a directory contains over 50 files, they consider splitting it.

The \UTIL directory is a good place to examine the simple-versus-complex directory issue.

You probably have a collection of handy, efficient software utilities. If you don't already, you soon will. Some are single stand-alone programs. Others may include files for configuration options, documentation, or on-line help. Some, such as the *Norton Utilities,* consist of many programs.

The issue is whether to lump all these program and support files into a single large \UTIL directory or to create subdirectories within \UTIL for separate programs.

A few programs, such as *FastBack* and *HomeBase,* give you no choice. They must go into their own level one directory. There is nothing to decide in such cases. But what will you do about the rest?

The advantages of a single large UTIL directory are:

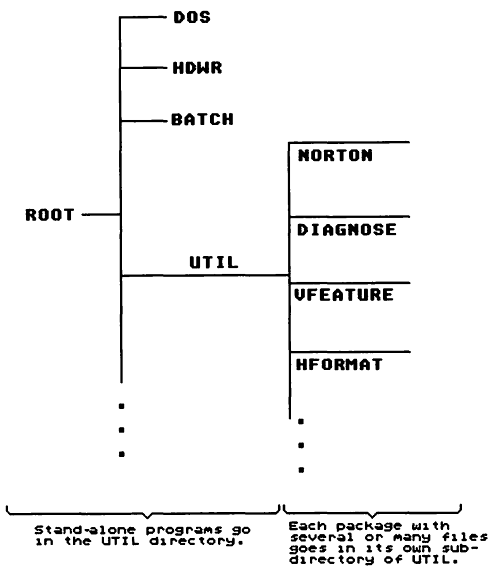
- The structure is simple.
- With the PATH command, discussed later in this chapter, all files in the \UTIL directory are available without batch files or other special processing.
- You don't waste disk space on directories, which take 2K or more.

The disadvantages of a single \UTIL directory are:

- Different utilities may share names given to common support files. For example, many utilities put last-minute instructions into a file called README. If you blindly copy many utilities into a single large directory, later files will overwrite earlier files of the same name.
- It's hard to tell what files belong to which utility program.
- It's hard to update utility programs because you don't know exactly which files to replace.
- You may cause fragmentation as you add more clusters to store all those directory entries.

Many experts recommend creating separate subdirectories for any utilities with more than a few files. You might make exceptions if a program's filenames are easy to recognize and not likely to be duplicated by another program. You'll need to create batch files or extend the search path, as will be discussed later, to access each program from its separate subdirectory. However, that's a small price to pay for a clean, easy-to-understand disk organization. Figure 4-7 depicts such an organization.

Figure 4-7. A Sample \UTIL Directory Organization



Another possibility is to make separate level one subdirectories (directly from the root directory) for each different utility, rather than level two subdirectories within \UTIL. However, this clutters the root, so we advise against it.

Backups

It's much easier to plan backups if all the data within a directory is similar. Data files are constantly changing, so they need regular backups. Programs remain the same once they have been installed, so they don't need frequent backup. If program files are in a separate directory, it's easy to forget and leave them out of the regular backup procedure.

Some data is more volatile than others. You may want to save your budget files weekly, but your production files daily. If so, put budget files in one directory and production in another. No matter what backup program you use, it's easier to back up an entire directory than to name individual files.

Security

If Joe needs one set of data and Sue needs another, consider giving them each a separate directory. If necessary, you can add password protection or encryption (see Chapter 7) to keep them out of each other's files.

But even if confidentiality isn't an issue, separate directories make sense. Joe is less likely to cause accidental damage to Sue's files if he has no reason ever to enter her directory.

Program Requirements

As mentioned above, some programs have their own requirements, which may not fit your overall plan. Here are a few examples:

- Some programs must go into level one directories.
- Others must go into subdirectories with a particular name. *RBase 5000* looks for programs in directory \RBASE, and in \DBFILES for data and help files.
- Some programs do not understand path names, such as *WordStar* version 3.3.
- Certain programs require a particular environment. *Software Carousel* won't work reliably with RAM-resident programs present.

Fortunately, PATH and related MS-DOS commands, plus well-designed batch files, can overcome nearly all program-related problems. Until you discover otherwise, assume that program requirements needn't thwart your disk organization plan. Now we'll discuss the PATH command and look briefly at batch files, so you'll have full information available before you make your final decisions.

Paths and the Environment Area

What happens if you're in your C: \WORD word processing directory and you want to run the MS-DOS BACKUP command, located in the C: \DOS directory? Without help, MS-DOS only recognizes files within the current directory. If you're using Version 3.0 or later of MS-DOS, you can execute C: \DOS \BACKUP. In earlier versions of MS-DOS, you'd have no choice but to leave \WORD and move to \DOS. No choice, that is, unless you used the PATH command. PATH and related MS-DOS commands offer an easy way to broaden MS-DOS's vision.

The PATH Command

The PATH command first appeared with MS-DOS Version 2.0. It lets you list any number of directories in which to search for executable program files. The syntax is:

PATH [drive]PATHNAME;[drive]PATHNAME; . . .

The directories can be on different drives and at different nesting levels. Notice that there are no blanks allowed between list entries. When you enter a program name without a complete path specification at the MS-DOS prompt, MS-DOS searches for the program as follows:

- In the current directory
- In the first directory named in the PATH command
- In the second directory named in the PATH command
- And so on

Therefore, it's important to list directories in order of frequency of use. If you're using \DOS, \HDWR, \UTIL, and \BATCH directories as recommended, you'll want all four in your PATH. A typical sequence would be

PATH . .;C: \BATCH;C: \HDWR;C: \UTIL;C: \DOS

You want to supercede generic MS-DOS programs in \DOS with any nifty versions you've put into \UTIL. But if you have a hardware-specific version of the same or similar program in \HDWR, you'll want to use it instead. If you went to the trouble of creating a smart batch file to control the process, you surely want the \BATCH file run instead of the program. Since you will often be in a data directory (\WS4 \LETTERS) that is the child of the application's directory (\WS4), you have the system first look in the parent of the current directory (. .).

Don't include the root directory in the PATH. After all, you shouldn't be keeping any program files there. Within a single directory, MS-DOS searches first for a file matching the specified program name and having the extension .COM, then for a matching program name with the extension .EXE, then .BAT. By using the PATH sequence described above, you can insure that batch files take precedence over similarly-named programs. Within the batch file, you must use full path specifications to avoid reinvoking the batch file.

You can include as many directories in the PATH as you like. Many people include the directories containing their major word processors, database managers, and other large, complex programs. However, long PATH lists can be bad for two reasons:

- The more directories, the more chance of accidentally encountering duplicate program names, and therefore of executing the wrong program.
- If you enter a bad program name, MS-DOS must search all those directories before telling you about your mistake.

Only one PATH command is in effect at a time. This means you must include all desired directories in each PATH command, as they will not carry over from previous PATHs (although in Appendix B you will see how to do this with batch files). To see what directories are currently part of the search path, type

PATH

with no parameters. You can cancel the search path completely by entering

PATH;

The APPEND Command

The PATH command only works with program files. MS-DOS version 3.2 offers a new command, *APPEND*, which extends the search path for all other files. PC-DOS 3.2 doesn't offer it; rather, it's included with network software. If you don't have APPEND, you can use ArborSoft's *SEARCH* (see Figure 4-8) or IBM's *File Facility*. APPEND lets you broaden the search for data files, program overlays, and support files such as on-line help files. Syntax is the same as for the PATH command:

APPEND [drive]PATHNAME;[drive]PATHNAME; . . .

Figure 4-8. A Substitute for APPEND

The SEARCH *program gives APPEND performance to MS-DOS 2.0 through 3.1.*

```
C:\UTIL>search /h

                          SEARCH 1.0

SEARCH is a general purpose replacement for the DOS PATH command.
SEARCH works with arbitrary programs on arbitrary files. SEARCH
is distributed on a user-supported basis. Type SEARCH /L for
licensing information.

Available functions are:

    Set....................... SEARCH [d:]path[[;[d:]path]...]
    Clear..................... SEARCH;
    Display................... SEARCH
    Activate.................. SEARCH /A
    Deactivate................ SEARCH /D
    Edit (with function keys). SEARCH /E
    Help...................... SEARCH /H
    License Information....... SEARCH /L

Copyright (C) 1984  Arborsoft Systems, Inc.
                    P.O. Box 4599
                    Ann Arbor, Michigan 48106
```

APPEND's functioning is parallel to PATH.

- You can have any number of directories in your APPEND list.
- MS-DOS searches the current directory, then APPENDs directories in the order listed.
- Each APPEND command replaces all previous commands.
- *APPEND;* cancels the APPEND search path.
- *APPEND* with no arguments displays the current search path.

APPEND's real value is with read-only overlay and support files. It can be dangerous with data files. Using APPEND, your program will find and read the data file just fine. However, it may not update it in the right directory. Therefore it is recommended that you give explicit pathnames for data files.

ASSIGN, SUBST, and JOIN

ASSIGN, SUBST, and *JOIN* are three high-powered, clever, but dangerous MS-DOS commands. They are important tools in creating a workable hard disk environment for all your temperamental software. But all have potential pitfalls. Use them carefully.

ASSIGN. ASSIGN was the first of the advanced tree commands. Introduced in MS-DOS version 2.0, it lets you substitute one disk drive for another.

In the days of systems which only had floppy disk drives, programs would frequently expect to find their support files on drive A and their data files on drive B. But if you had a PC XT, you had no floppy drive B and the program files were on your hard disk, drive C. You could tell the program to look for the support files on drive C rather than A and the data files on drive A instead of B with this simple command:

ASSIGN B = A A = C

If you wanted to cancel the drive reassignment, you could do so by typing ASSIGN with no parameters.

Warning. Use ASSIGN with care.

- Do not change versions of MS-DOS while using ASSIGN. In particular, do not boot from floppy disks that may contain an earlier version of MS-DOS. You may find yourself stuck, unable to return to the original drive assignments.
- Cancel the ASSIGN immediately when you're done with it. Otherwise, you may find yourself using, changing, or removing data from the wrong drive.

SUBST and JOIN. SUBST and JOIN are external commands in MS-DOS Version 3.1 and later. They fool your programs into thinking directories are drives and vice versa. SUBST lets you substitute a drive identifier for a directory. From that point on, the directory appears like a drive to your programs. This is extremely useful for programs like *WordStar* 3.3, which can work with alternate drives but not with alternate directories. The syntax is

SUBST ***drive: directory***

Until revoked, **SUBST E: C:\WS\MEMOS** causes MS-DOS to act as though all files in the C:\WS\MEMOS directory are actually on the current directory of disk drive E. Notice that an absolute path was specified for the directory name. Relative paths may cause problems.

Never use a real disk drive for your substitution. You can use any nonexistent drive through drive E. You can also use drives F through Z if you add a LASTDRIVE command to your CONFIG.SYS file. CONFIG.SYS and the LASTDRIVE command will be discussed later in this chapter.

JOIN lets you treat disk drives as though they were subdirectories. Thus, you can treat a RAM disk, a floppy disk, or a second hard disk as part of your hard disk. The format is:

JOIN ***guest-drive host-drive\pseudo-directory***

For example, suppose you have data on drive A. You are using a program on drive C, and most of your data is on C. The program requires all data to be on the same drive. You could use a JOIN command, **JOIN A: C:\dummya** to fool the program into thinking all data was on C. While very few programs recognize subdirectories but not multiple drives, it's good to know the possibilities.

JOIN requires these conditions:

- Neither host nor guest drives can be part of a current ASSIGN or JOIN.
- Neither the guest drive nor the host subdirectory can be current or part of a network.
- The host subdirectory must be a first level directory (a child of the root), and it must be empty.

If the host subdirectory doesn't exist, MS-DOS will create it. Both SUBST and JOIN display their current status if you enter the command with no parameters.

To cancel a SUBST or JOIN, type the command, the drive identifier, and */D* for delete. For example, to cancel a JOIN in which drive A is the guest drive, type **JOIN A: /D**. To cancel a substitution of drive E, type **SUBST E: /D**. ASSIGN, SUBST, and JOIN offer great power and flexibility. With power comes danger, however. All three increase the possibilities for mix-ups, accidental changes or deletions, or worse.

Use ASSIGN, JOIN, and SUBST conservatively and with care.

- Don't use any of these commands if a less drastic solution is available.
- Cancel the command as soon as its work is done.
- Never use more than one of these three commands at the same time.
- When any of these commands are in effect, do not use the following DOS commands:
  FDISK
  CHKDSK
  DISKCOPY
  FORMAT
  BACKUP
  RESTORE

Introduction to Batch Files

You've seen the MS-DOS commands PATH, ASSIGN, JOIN, SUBST, and PROMPT. You understand the directory commands MD, CD, and RD. You can type these individually at the MS-DOS command prompt. However, all that typing can become tiresome,

especially if you find yourself repeating the same command sequences over and over.

This is where batch files come in. In essence, a batch file is a collection, or *batch*, of MS-DOS commands. Rather than executing as you type them, they execute one after another from the batch file.

You can create a batch file using any editor which creates ASCII text files. The file must contain only printable characters. Many word processors add special control characters which are unacceptable. However, most word processors have a nondocument mode which creates clean ASCII text. EDLIN, the MS-DOS line editor, is clumsy, but it does create acceptable ASCII files. A final possibility is using the COPY CON command, which copies everything you type on the keyboard into a file.

Batch files all have the same extension: *.BAT*. To execute a batch file in the current directory, simply type its name when you see the MS-DOS command prompt. Essentially, the batch filename acts like a new MS-DOS command. MS-DOS finds the batch file, reads it, and executes its contents one line at a time.

If you provide path information in the file specification or via a PATH command, MS-DOS can execute a batch file from another directory and/or drive.

AUTOEXEC.BAT

When you power up or reboot your computer, MS-DOS runs through a formal startup procedure. As the last startup step, MS-DOS searches the root directory of the drive from which it loaded the operating system. It looks for the startup batch file, AUTOEXEC.BAT. If it finds it, MS-DOS executes its contents.

AUTOEXEC.BAT is optional but should be considered required for hard disks. It is recommended that your AUTOEXEC.BAT file always contain a single line:

C:\BATCH\STARTUP

This way, whenever some software installation procedure obliterates your AUTOEXEC.BAT file, you can easily see what was added. The STARTUP.BAT file within your \BATCH directory then is your real startup file.

In it you set the date and time, initialize your serial ports,

change the MS-DOS prompt, define an overall PATH, start any RAM-resident software, and make always-needed tweaks to your system.

Figure 4-9 shows a minimal STARTUP.BAT. Let's examine it. The first two lines issue DATE and TIME commands. If MS-DOS doesn't find AUTOEXEC.BAT at startup, it issues DATE and TIME commands itself. Therefore, you are probably familiar with these prompts already, even if you didn't know what they were. If AUTOEXEC.BAT is present, MS-DOS doesn't issue DATE and TIME; you must do so yourself. The PC AT contains a calendar and clock, as do PC systems with add-on boards such as the AST Six-Pak. If your system contains a realtime clock, you may want to run whatever software in your \HDWR directory links the MS-DOS date and time to the hardware clock, such as *ASTCLOCK*.

Line three sets the PATH to include the four basic support directories. As discussed earlier in the chapter, this lets you run programs from these directories regardless of the current drive and directory.

Line four causes MS-DOS to verify the proper recording of written data. MS-DOS does this by reading each sector after writing and comparing the memory-buffer original with the disk copy.

Line five changes the MS-DOS command prompt to display the current drive and directory.

Figure 4-9. A Minimal \BATCH\STARTUP.BAT File

```
DATE
TIME
PATH . .;C:\BATCH;C:\HDWR;C:\UTIL;C:\DOS
VERIFY ON
PROMPT $P$G
```

If you already have an AUTOEXEC.BAT file, turn it into your STARTUP.BAT file in the \BATCH directory with these commands:

```
COPY C:\AUTOEXEC.BAT C:\BATCH\STARTUP.BAT /V
DEL C:\AUTOEXEC.BAT
```

If you had no AUTOEXEC.BAT to turn into \BATCH\STARTUP.BAT, use the COPY CONsole command to enter the one from Figure 4-9.

- From the MS-DOS prompt, type: **COPY CON C: \BATCH \STARTUP.BAT**; then press Enter.
- Carefully type in each line from Figure 4-9.
- If you have an IBM PC AT or compatible, or any other system with a built-in clock, don't type the DATE and TIME lines; you don't need them.
- Press Enter after each line.
- Press F6 or CTRL-Z after the last line.

Now create your AUTOEXEC.BAT with the COPY CONsole command.

- From the MS-DOS command prompt, type **COPY CON C: \AUTOEXEC.BAT**; then press Enter.
- Now type **C: \BATCH \STARTUP.BAT**; then press the Enter and F6 (or CTRL-Z) keys.

To test your STARTUP and AUTOEXEC files, restart your system. Press the reset button (if you have one), or hold down the <CTRL>, <ALT>, and <DEL> keys at the same time.

Some Simple Batch Files

You've learned about directories, directory-related DOS prompts, and batch files. The best way to put it all together is to show a few examples. Then it'll be your turn to finish designing your directory structure, create your directories and load up your disk.

If you set up your disk well, every major software package you use will be in its own directory. The data files for each program will be in one or more directories, which may or may not be subdirectories of the program directory.

How will you locate the program and start it? How will you insure that it can find and read its own overlay and support files, and find your data files?

The answer is *batch files*. Every type of processing performed repeatedly should have a corresponding batch file in your \BATCH directory.

In the simplest case, a program plus all its support files, overlays, and data are in a single directory. All you need to do is change to that directory and execute. Let's assume that the program is the public-domain communications program *MSKERMIT*, in a directory called \KERMIT, a subdirectory of D: \COMM. Create

batch file KERMIT.BAT, as listed in Figure 4-10, and put it into the C: \BATCH directory. C: \BATCH is part of the PATH statement executed in your STARTUP.BAT. Therefore, typing KERMIT at the MS-DOS command prompt will run C: \BATCH \KERMIT.BAT, no matter what directory you're in. It will bring you into the D: \COMM \KERMIT directory and start the MSKERMIT program.

Figure 4-10. A Sample Batch File, TEST.BAT

```
D:
CD D:\COMM\KERMIT
MSKERMIT
C:
CD C:\
```

Notice the last two lines of Figure 4-10. They log into the root directory of the first hard disk after finishing the program. It's good practice to leave things as you found them. Batch files should undo any special changes they initiate.

As mentioned earlier, *WordStar* 3.3 does not recognize subdirectories. However, it does recognize different drives. The batch file in Figure 4-11 uses the SUBST command to fool *WordStar* into thinking the data files it needs are on a different drive, not just a different directory.

Figure 4-11. Fooling *WordStar* 3.3 with SUBST

```
C:
CD C:\WS
SUBST E: C:\WS\REPORTS
WS
CD C:\
SUBST E: /D
```

First, the batch file logs onto the hard disk, C, just in case you were previously using a floppy disk or another hard disk drive. Then it changes to the \WS directory. This enables *WordStar* to find its overlays and support files.

Next, it performs a substitution, fooling *WordStar* into thinking that the *WordStar* report data directory, C: \WS \REPORTS, is another drive, E. Having completed preparations, you're ready in the fourth line to start running *WordStar*. Once you exit *WordStar*, the batch file continues.

The last two lines don't execute until after *WordStar* ends. These lines reestablish the root as the current directory and cancel the SUBST command.

In Figure 4-11, the only subdirectory allowed for was \WS \REPORTS. But \REPORTS may be only one of several data directories for *WordStar*. When one program has several data directories, either you will want a separate batch file for each, or you'll use a *parameter* to define the directory. Modify the WS.BAT file, replacing the subdirectory *REPORTS* with *%1*, as in Figure 4-12. Then run the batch file by entering the command WS REPORTS or WS MEMOS to use directory \WS \REPORTS or \WS \MEMOS.

Figure 4-12. Modifying WS.BAT

```
C:
CD C:\WS
SUBST E: C:\WS\%1
WS
CD \
SUBST E: /D
```

This section has given the barest hint of the potential use of batch files. In the next chapter, the usefulness of batch files in creating integrated job streams will be discussed. Also, Appendix B introduces the entire programming language available for processing parameters, decision-making, and branching within batch files. It is recommended reading.

Boost Performance with CONFIG.SYS

A simple way to customize your hard disk system is by using *CONFIG.SYS*. CONFIG.SYS is a small file, typically a few hundred bytes or less. DOS reads it as part of the system startup procedure. It tells DOS how to tailor the operating environment to your special needs. You don't have to have a CONFIG.SYS file. Without it, DOS simply uses a standard (default) operating environment. However, creating even a few lines for CONFIG.SYS can significantly improve performance.

Increasing the Number of Buffers

MS-DOS doesn't read or write data directly between your disk and its ultimate memory location. Instead, it moves it through buffers. DOS buffers are 512-byte chunks of RAM. Whenever MS-DOS receives a read or write request, it first checks whether the requested data is already in a buffer. If so, MS-DOS gets the data directly from the buffer without accessing the disk. Since RAM access is roughly 100 times faster than disk access, this saves much time.

By default, MS-DOS uses three buffers for a hard disk system. It uses two for a floppy disk–based system. These numbers are far below optimum. The ideal number depends on the kind of disk accesses you typically need—short, random reads/writes, or long, sequential ones. However, the best number is usually between 10 and 40 buffers. Too many buffers tie up precious RAM; too few force your computer to make more of those slow disk accesses.

The BUFFERS command in CONFIG.SYS lets you specify from 1 to 99 buffers. The format is:

BUFFERS = *number*

Try using 20 buffers. If your system seems sluggish, you may want to experiment later with other values. Unfortunately, there is no systematic way to determine the ideal number of buffers. What would seem the perfect number to one person would not be right for someone else.

Increasing the Number of Files

Unless told otherwise, MS-DOS Version 2 allows you to have five files open at a time. MS-DOS Version 3 allows eight. Many sophisticated programs use far more. For example, a word processor may have several program files and overlays. It may have open at the same time—a thesaurus, spelling files, printer drivers, help files, backup files, and temporary work files, as well as several data files all open at once.

The FILES command lets you have up to 20 files open at once for MS-DOS version 3.2 and earlier, 255 for version 3.3. The format is:

FILES = *number*

Adding Device Drivers

Device drivers were first mentioned in Chapter 3. Device drivers tell MS-DOS how to handle nonstandard or extra-large hard disks. Device drivers in general give MS-DOS special instructions on working with any nonstandard piece of equipment, including a mouse, expanded memory board, or a nonstandard printer.

Each device driver is a file on disk. MS-DOS is informed of its existence and location by one line in CONFIG.SYS for each device driver:

DEVICE = [pathname] \filename [optional parameters]

ANSI.SYS is a special device driver. It's provided free with MS-DOS Version 2 and later. ANSI.SYS handles video display colors and graphics and keyboard output. Many installations include it within CONFIG.SYS. Don't. It just takes up memory, and it provides little benefit unless you have software that specifically needs it.

If you must use the features of ANSI.SYS, consider the Fansi-Console *FANSI.SYS* instead. This device driver is much faster than the MS-DOS version, and it provides additional features.

Don't put device drivers in the root directory. Keep the root clear. Put device drivers in the \HDWR directory. The only penalty for keeping drivers out of the root is that you must specify the full path to the device driver. For example, to use the standard ANSI.SYS device driver from the \DOS directory, enter

DEVICE=C:\DOS\ANSI.SYS

MS-DOS Version 3 and later provides another free device driver, VDISK.SYS. This treats a portion of RAM as an extremely fast *virtual* disk. RAM disks will be discussed in Chapter 6.

Setting the Last Drive Identifier

When the SUBST command was explained, you saw why it can be important to fool MS-DOS into thinking you have many disk drives. Drive identifiers A and B are usually reserved for floppy disk drives. If you only have one floppy disk drive, MS-DOS views it as both A and B to allow floppy-to-floppy disk copying. Your first hard disk is usually C. A second hard disk is D, then E, and so on. If you'll be using MS-DOS commands to create pseudo drives beyond E, you'll need a LASTDRIVE in CONFIG.SYS. The format is:

LASTDRIVE = ***letter***

The letter ranges from *F* to Z. Unlike the previously discussed CONFIG.SYS commands, the LASTDRIVE command doesn't increase the size of MS-DOS.

Other Options

Your DOS reference manual discusses other CONFIG.SYS commands. These include:

- COUNTRY: Selects the format for dates, times, currency, and separators. You can choose from 15 formats ranging from United States to Australia, Finland, and Israel.
- BREAK: Tells whether you can interrupt programs or commands any time or only during input/output.
- SHELL: Describes the command processor and the size of the environment area. This lets you move COMMAND.COM from the root directory or use your own shell program instead.
- STACKS: Gives the number of stacks and the size of each.

Creating CONFIG.SYS

CONFIG.SYS is an ASCII text file. It contains letters, numbers, and a few special characters such as the colon (:) and backslash (\ ). It must not contain the special, hidden codes that many word processors add to document files. Just as with batch files, you make CONFIG.SYS by using your word processor in ASCII or nondocument mode. Or you can use the COPY CONsole command, abbreviated COPY CON.

To use the COPY CONsole command to create a minimal CONFIG.SYS:

- From the MS-DOS prompt, type **COPY CON C:\CONFIG.SYS**; then press Enter.
- Type **BUFFERS=20**; then press Enter.
- Type **FILES=20**; then press Enter.
- Press F6 or CTRL-Z to end the file.

Once you have created CONFIG.SYS, save a copy of it in the \BATCH directory with the command

COPY C:\CONFIG.SYS C:\BATCH /V

Set Up Your Directories with Most-Used Items First

If you haven't done so already, stop now and map out your hard disk organization. Sketch a tree similar to Figure 4-6.

The structure will change as you add new programs, applications, projects and personnel. However, make it as complete as you can at this point. Include all directories you know you'll need within three to six months. Check to see that you've used simple, meaningful directory names. Once you've completed your fundamental structural decisions, you're ready to create your directories and start loading up your disk. It's important to do these two steps now and to do them systematically. To understand why, it's necessary to review how MS-DOS reads, writes, and locates disk files.

Most hard disk drives position themselves from cylinder zero, the outermost cylinder. Thus, even if a directory or file is on the very last cylinder, the drive heads start at the first cylinder and step out. Furthermore, if a file is fragmented, meaning it is on noncontiguous clusters, the heads must reposition, starting from cylinder zero, before reading each new noncontiguous cluster.

Before MS-DOS reads any file, it goes through these steps:

- It goes to the specified directory, or the current directory if none is named.
- It reads sequentially through the directory entries, looking for the specified filename. It uses the directory information to find the starting cluster record for this file in the FAT.

From this information, it's clear that

- Frequently used files should be recorded on the hard disk as close to cylinder zero as possible.
- Directories are the most frequently used files, because there is at least one directory access for every file access.
- Within every directory, putting the most frequently used files first will save directory search time.
- Contiguous files process much faster than noncontiguous files.

Ideally, we want our files on the hard disk in the following sequence:

- Most frequently used directories
- Other, less frequently used directories

- Most frequently used and least changing files (usually program and support files)
- Other files

Remember that:

- Files within each directory should start with the most frequently used.
- Files most likely to fragment, through frequent size changes, should go after other files, if possible. If they're ahead of other files, they're likely to cause fragmentation of these files too, as fragmented clusters become available within the FAT.

How do you accomplish all this? Roughly speaking, the first files you put onto a disk go to the best locations, closest to cylinder zero. This is because of the way MS-DOS uses the FAT. When you write a file, MS-DOS searches the FAT from the beginning for the first available space, or cluster. If the disk is empty, the first cluster available will be on cylinder one, immediately after all system files and the root directory. Hence, the keys to physical disk layout are frequency and stability. Planning pays in performance.

Conclusion

This chapter has covered the basics of tailoring a hard disk system to your needs. You've learned about directories, paths, batch files, and CONFIG.SYS. You're aware of some of the many factors involved in setting up a good directory tree. By thinking now about your needs, you can save many hours of waste and frustration in the future.

In the next chapter, you'll look at more tools to help make a fast, easy-to-use hard disk system. More MS-DOS commands will be introduced, and there will be more about batch files. Then you'll go beyond MS-DOS to the many support utilities available for easy file handling.

Files should be recorded on your disk based on frequency of use and stability.

Load your hard disk systematically, *after* you plan.

- Create the \DOS, \UTIL, \HDWR, and \BATCH directories.
- Create your other first level subdirectories.
- Create as many other second- and third-level subdirectories as you think you may need.
- Copy MS-DOS into the \DOS directory.
- Set up AUTOEXEC.BAT, and minimal STARTUP.BAT and CONFIG.SYS.
- Reboot and use your new PATH and support directories as you finish loading the hard disk.
- Copy hardware-support programs into the \HDWR directory. If you need to use device drivers, add them into CONFIG.SYS and then reboot.
- Copy your system utility programs into the \UTIL directory. If you want to include them in your STARTUP.BAT, do so now and then reboot.
- Copy application programs and support files into their directories. In general, program files don't change, and they are used often. It makes sense to put them near the start of your disk. Use the following sequence:
  1. Load overlay files first, as they get the most use. These usually have extensions of .OVL, .OVR, or .PRG.
  2. Load the most frequently used program and support files.
  3. Load the remaining program and support files.
- Run any customization or installation procedures.
- Back up *all* files. This becomes your worst-case backup. It insures that all your programs are saved somewhere. See Chapter 8 for details.
- Load alterable data files into the appropriate data directories.
- Perform an *incremental* backup. This will save all your newly-loaded data files. See Chapter 8 for details.

Chapter 5

Usage Tips

Usage Tips

When you buy a new car and bring it home, all shiny and new, it's hard to imagine that it won't look like that forever. It won't. Yet, with a routine of weekly washings and quarterly tuneups, your car will serve you for years.

Your hard disk will serve you for years, too, with a minimum of maintenance. This chapter will help make life with a hard disk simple, satisfying, and long.

You'll learn some management techniques to keep your disk cleanly organized and easy to use. Then you'll have a look at various tools to help you keep organized.

> Hard disks need small doses of efficiency, applied consistently.
>
> - Keep your disk uncluttered.
> - Use consistent file names.
> - Keep directories small.
> - Remove unneeded files.
> - Use a \SCRATCH directory.
> - Keep data on floppies.
> - Get maximum benefit from directory listings.
> - Use utilities to help identify files.
> - Compress or combine data where appropriate.
> - Run job streams, not just programs.
> - Unify your operating interface with
>   - Menu programs
>   - Macro processors
>   - Environment managers
> - Treat multiple systems as generic software sites.

Running a Clean Ship

The first rule of hard disk use is to keep your disk organized. In Chapter 4, you designed your basic rules of directory organization. Now your job is to stick to them. These rules are important because:

- Hard disks can hold hundreds or thousands of files. Lack of consistency in naming and locating directories and files will waste hours of your time, hunting.
- Moving files takes time and creates holes which fragment your disk. Put files in the right place the first time, unless you sort your directories (see Chapter 6).
- Not knowing what's what increases wasteful clutter. You're more likely to leave extra, unneeded files on disk. Your hard disk will fill up faster. There will be more for both you and MS-DOS to search through.

With these points in mind, let us look at some basic rules of hard disk housekeeping.

Use Consistent Names

As discussed in Chapter 4, you should use extensions for all file names, but not for directories. If you sort your directory entries by extension, as in Figure 5-1, then subdirectories appear first and speed up all of MS-DOS' file searching.

Give all related files consistent names, with identical length and structure. Don't put important identifying information in file-name extensions; many programs change these to .BAK in backup files.

Keep Directories Small

The MS-DOS DIR command displays 20 directory entries per screen. Sixty-four entries fit in a 2048-byte cluster. Somewhere between these two numbers is the maximum number of files you should keep in one directory.

Group files logically by content, function, or personnel. Whenever directories become unwieldy, split them. If needed, use SUBST, ASSIGN, JOIN, (see Chapter 4) or a commercial file finder (discussed in the Chapter 6) to help your programs locate data files.

Figure 5-1. Sorting a Directory by Extension

The Mace+Utilities *lets you sort directories by filename extension to put subdirectories first.*

```
                MACE+UTILITIES 20M/256K
           Ver 3.21 Copyright 1986, Paul Mace
                  All rights reserved.

 F1--HELP                      F2--OTHER UTILITIES
 F3->Diagnose                  F4--CHKDSK
 F5->Remedy                    F6->Squeeze/Sort DIR's
 F7->Condense (UnFragment)     F8->Create BACKUP.M_U

              Select one!   <Esc> to exit.

        Directories on selected disk will be condensed.
           You will have the option of sorting by
         Name, Extension, Date-time or Size of file.
```

Store Data and Old Files on Floppy Disks

The spark plugs become fouled if you use your car only for short trips to the corner store and back. Likewise, your hard disk will get sluggish if you overload it with short-term files.

Analyze every file to see if it belongs on your hard disk. If not, either delete it or move it to a floppy disk. Label the floppy disks clearly and segregate them by content. Consider using different colored floppy disks, or at least different colored labels, for each application. Store the floppy disks within their dust jackets in a safe place. Protect them from spills, creases, and magnetism.

The importance of protecting disks was brought home to the experts early in 1985. Fujitsu Systems of America completed a move into new corporate headquarters in Del Mar, California. They occupied modern new buildings, replete with modular furniture workstations for clerical and software development personnel. Over the ensuing few months, some staff found that both new and old disks would go bad, literally overnight.

They would put good disks into the overhead cabinet in their cubicles and go home. When they returned to work the next morning, they would find they were unable to read the data on their disks.

It turned out that the errors were introduced by turning on the cubicle's lights, recessed within the underside of the overhead cabinets. Each fluorescent light had a starter that generated enough magnetism to affect disks directly above, upon the cabinet shelf.

Many word processors automatically create online backup files with the extension .BAK. When you do your regular backups (see Chapter 8), remove all .BAK files from the hard disk.

When you finish a project, archive its data files to floppy disks. Label it well and store in a safe place. If it's extremely important data, make two copies and store them in separate locations. Then remove the files from your hard disk.

Because *newer* is not always *better*, it's often wise to retain the old version of a newly-updated program. Sometimes you will make a special version of a program, installing options that make occasional jobs easier. Consider keeping the superceded or less-used version of the program on floppy disks.

Even current data files may belong on floppy disks. The best candidates are:

- Rarely used files
- Small files that will fit entirely into memory
- Files that do not need to be integrated with other files

Letters and other small word processing files are good examples. You may have dozens or even hundreds of such files. They clutter your disk unnecessarily. Once you're done with them, get them off of your hard disk.

As with archived files, store data files logically. Think of each floppy disk as a subdirectory. Identify both its parent application or project and the specific contents of this disk. Don't load floppy disks randomly, just filling them up. If you do, you'll end up sorting through 30 disks to find three related files.

Keeping data on floppy disks can also be a good system if several people are sharing one computer system. There's no danger of harming each other's files, and your data is always easy to find.

Use a \SCRATCH Directory for Experiments

Everyone should make a directory named \SCRATCH. Use it for anything you know is temporary:

- Software you want to load up just long enough to look at
- Test files
- Brief experiments
- Intermediate work files

Using a temporary directory in this way provides two benefits. You needn't worry about inadvertantly overlaying a production file or already-installed program. Also, you avoid cluttering your disk with needles in the haystack, short-term files buried in the midst of permanent data.

Make a habit of clearing out the \SCRATCH directory regularly, perhaps via your startup batch file. This minimizes the risk of keeping old files just because you're not sure what they are. If it's in the \SCRATCH directory, you know you can remove it. Of course, there's a risk here. If you put something in \SCRATCH and later decide to keep it, you'd better move it to another directory.

Get Help Keeping Track of Files

The programs described in this section have one main purpose: to help you identify all your many files. Three main kinds of help are available:

- Enhanced directory listings (DIR, SD)
- Programs to add notes to files and directory listings (*PC Easy II, Zoo Keeper, SmartNotes, DayFlo Tracker*)
- Text searching programs (*Norton Utilities, Dragnet, ZyIndex*)

Maximum mileage from directory listings. The DIR command was introduced in the last chapter. DIR lists the contents of the current directory. Before looking at replacements, take a minute to learn two available options within DIR:

- /P pauses after each screenful of directory entries. The listing halts with the prompt *Press any key to continue. . . .* Pressing a key displays the next screenful of entries.
- /W (wide) lists only filenames, five per line. Thus it can display roughly one hundred filenames at once. Figure 5-2 shows a wide listing.

Figure 5-2. Wide Format Display with DIR/W

```
C:\BATCH>dir /w

 Volume in drive C has no label
 Directory of  C:\BATCH

.                ..               RETURN   BAT    FKEY10   BAT    UP       BAT
DOWN     BAT    OVER     BAT    SDIR     BAT    DIRS     BAT    TOUCH    BAT
YESNO    BAT    TLINK    BAT    CWD      BAT    PAINT    BAT    COPYDISK BAT
BACKUP   BAT    LIST     BAT    SAYWHERE BAT    ALPHA    BAT    WS       BAT
ERA      BAT    DOSTSR   BAT    DV       BAT    TCLINK   BAT    USEDRAFT BAT
VENTURA  BAT    TC       BAT    SET-CWD  BAT    GOTO     BAT    GEM      BAT
ERADIR   BAT    USELASER BAT    ROOT     BAT    VP       BAT    ALLTSR   BAT
NUMKEY   BAT    PROTECT  BAT    UNPROTEC BAT    APPTSR   BAT    HGCSNAP  BAT
CLR      BAT    MACE     BAT    STARTUP  BAT    CPM80    BAT    ALLDIRS  BAT
SET-CWD2 INP    SET-CWD1 INP    NUMKEY   MNU    FKEY10   MNU    DTG      BAT
CONFIG   SYS    KERMIT   BAT
       52 File(s)  1718272 bytes free
```

You can combine options. DIR /W/P will list 100 filenames, then will pause until you press a key. Many other programs offer far more useful directory listings than DIR. Most provide these features as part of a more complete file manager. These will be discussed in Chapter 6. However, the public domain program SD.COM (Sorted Directory) provides a better directory-listing than DIR. It would be a good idea to rename this program as D.COM (Directory), to avoid filename conflicts with *Norton*'s SD.EXE. (The *Norton* SD program sorts the actual directory file, as does the *Mace* utility in Figure 5-1.) How is SD.COM better than DIR? First, SD uses a two-column format, displaying twice the information per screen. Second, it offers numerous options for displaying entries, as listed in Figure 5-3.

Various file manager programs, or shells, offer additional directory features. These will be examined in Chapter 6.

Making notes on directory entries. MS-DOS filenames are a mere eight characters long, plus a three-character extension. It's hard to meaningfully identify the contents of hundreds of files from these cryptic names. Software developers have produced several levels of additional file identification, including:

- Longer filenames
- *Paste-on* notes and comments
- Key word identifiers

Figure 5-3. The public-domain SD (Sorted Directory) program offers many options not available with DIR.

```
PC-DOS SD V4.3 (c) by John Stetson 14-Oct-84

Usage: sd [dev:][path][fspec] /b/d/e/n /c/q/r/s /l/t/v/x

[dev:]  - disk device for directory display
[path]  - path to the desired disk directory
[fspec] - file name pattern specification to match

/b - sort file entries by file sizes in bytes
/d - sort file entries by modification date & time
/e - sort file entries by file name extension
/n - display entries in directory order (no sort)

/c - clear console screen before displaying output
/q - quick output (no pauses between pages)
/r - produce output suitable for redirection
/s - include system and hidden files in output

/l - long output format (with file attributes)
/t - display the totals output line only
/v - display volume and directory name info
/x - display the command line syntax summary

C:\>
```

Longer filenames: PCEasy II. PCEasy II is an $85 utility with one purpose: it allows longer, more meaningful filenames. Names can be up to 32 characters long, with upper- and lowercase letters, spaces, and punctuation.

Actually, *PCEasy II* is another form of note program. Despite appearances, MS-DOS filenames continue to be eight characters. *PCEasy II* attaches a longer synonym to the underlying name, then presents a shell for you to work with. You can sort directories, view files, and move from one directory to another. When you're running a program and need to select a file, you call a pop-up *PCEasy II* menu. This interface lets you select files using their extended names. It then returns the actual MS-DOS name to the working program. Interfaces are available for *Lotus 1-2-3, Microsoft Word,* and *WordPerfect*.

Paste-on notes. Several memory-resident pop-up accessory programs let you attach notes to your files. Different approaches are used by *DayFlo Tracker, SmartNotes,* and *Polaris Rescue.*

Dayflo Tracker locates files based on three criteria:

- The date you originally created the file
- The date you last updated the file
- User comments you attach to the file

You list which directories and files you want *Dayflo Tracker* to include. Whenever you close such a file, you will be prompted for your comments. The program then stores file access time stamps and your comments to help in identifying the file later.

SmartNotes, by Personics Corporation, takes a different approach. SmartNotes creates stick-on notes you attach almost anywhere. They can go into word processing files, spreadsheets, databases, or even other RAM-resident program files. As shown in Figure 5-4, they can also attach to directory listings.

Figure 5-4. Using Stick-on Notes

SmartNotes *can attach comments to directory entries, in addition to annotating the contents of data files such as memos and spreadsheets.*

```
C:\BATCH>d
.                 <Dir>  13-Aug-87  10:59 ¦ MACE     .BAT      43  26-Aug-87  15:27
..                <Dir>  13-Aug-87  10:59 ¦ NUMKEY   .BAT                         6
ALLDIRS .BAT         93  26-Aug-87  16:43 ¦ NUMKEY   .MNU Prototype batch menu    7
ALLTSR  .BAT         39  13-Aug-87  18:59 ¦ OVER     .BAT using number keys 0-9   5
ALPHA   .BAT        625  12-Aug-87  11:27 ¦ PAINT    .BAT (for disk book)         3
APPTSR  .BAT         78  16-Aug-87  21:55 ¦ PROTECT  .BAT                         4
BACKUP  .BAT        548  19-Jun-87  18:34 ¦ RETURN   .BAT                         9
BATCH   .DSN       4402  27-Aug-87  12:30 ¦ ROOT     .BAT                         5
CLR     .BAT         55  24-Aug-87  19:13 ¦ SAYWHERE.BAT       43  12-Aug-85  13:46
CONFIG  .SYS                             0 ¦ SDIR     .BAT      72  11-Aug-87  19:04
COPYDISK.BAT backup of version in        1 ¦ SET-CWD  .BAT                         6
CPM80   .BAT root directory              3 ¦ SET-CWD1.INP Creates CWD.BAT to       0
CWD     .BAT                             6 ¦ SET-CWD2.INP return to current drive 8
DIRS    .BAT                             3 ¦ STARTUP  .BAT and directory - lasts   3
DIRS    .SN                              6 ¦ TC       .BAT past power-off unlike   7
DOSTSR  .BAT                             7 ¦ TCLINK   .BAT PUSHDIR & POPDIR        7
DOWN    .BAT        186  21-Aug-87  16:12 ¦ TLINK    .BAT                         2
DTG     .BAT         96  27-Aug-87  09:49 ¦ TOUCH    .BAT     107  12-Aug-87  14:43
DV      .BAT         75  13-Aug-87  18:56 ¦ UNPROTEC.BAT        9  16-Aug-87  16:54
ERA     .BAT          8  29-Jun-87  23:59 ¦ UP       .BAT      60  21-Aug-87  16:11
ERADIR  .BAT        204  22-Aug-87  15:47 ¦ USEDRAFT.BAT
FKEY10  .BAT                              ¦ USELASER.BAT                          ON
FKEY10  .MNU  Prototype batch menu        ¦ VENTURA  .BAT selects LaserWriterPlus
SN: F1 Help  F2 Make-Note (dirs.sn)        F6 Defaults  F10 File  ESC Exit  Mode=T
```

Whenever you list a noted directory, its notes can be made to appear as well. You control whether notes appear automatically. You can erase notes or temporarily hide them. Since the notes reside in separate files, they don't alter actual file contents. Therefore noted directories work with other software as usual. The *SmartNotes* program uses 90K of memory and costs $80.

Polaris Rescue creates and manages online help files. These are similar to *SmartNotes* in design. Help screens appear when you press a user-defined help key. You can import regular ASCII text files into *Polaris Rescue* and can create networks of up to 1000 screens per *Rescue* file, with each screen pointing to as many as 30 other screens.

Identifying key words. Zoo Keeper, from Polaris Software, offers a handy variation on other directory utilities. The RAM-resident program pops up automatically whenever you create a file. It asks you for three key words and a 40-character description. When you want to find a file, enter part or all of one or more key words. *Zoo Keeper* lists all files in all directories meeting your criteria, along with their 40-character descriptions. You then select the file you want.

Searching for text and cataloging text files. Any of the preceding programs can save countless hours of search time. If you don't have such a program, though, a good text search tool is an important fallback.

Norton's TS program is an example of the simplest kind of text searcher. Part of the *Norton Utilities,* it hunts for a text string in any of three domains specified, including:

- An entire disk
- A file or group of files, with or without subdirectory traversal
- Erased files, that is, in disk areas not currently attached to a file

Figure 5-5 illustrates that TS displays the context in which the highlighted target text was found. TS is not case-sensitive; either upper- or lowercase text will match your input string: *Ca* matches either *CA* or *ca*. You can send the list of matches to a file or printer.

Figure 5-5. Highlighted Text

When Norton*'s TS (TextSearch) program finds the target text, it displays its context and location within the file.*

```
Searching D:\DISKBOOK\chap06.ws

Found at line 382, file offset 17,473

Computer Systems.  It costs around $600, but may be worth it if you're ì
serious about using expanded memory.


@NT SUBHEAD = Creating a RAM Disk @RO

You use special software to create a RAM disk.  This can be either
* a device driver loaded by your CONFIG.SYS file or
* a separate software program you run from the DOS command line

You can buy standalone RAM disk software.  Also, most expanded or extended ì
memory boards include RAM disk software designed to work with the board.  ì

Search for more (Y/N) ?
```

ZyLAB's *ZyINDEX Professional* goes further. It creates and searches text indices, like an automatic version of *Zoo Keeper*'s key word file. This makes it much faster than TS. However, you must reindex periodically to keep the indices current. A similar program is O'Neill Software's *Electra-Find.*

Dragnet is another text-search tool. It can use or ignore case information and accept partial matches. *Dragnet* allows combining multiple search criteria with logical ANDs and ORs. That is, if you specify that it should search for string one AND string two, it will find only occurrences where both strings occur together. If you specify string one OR sting two, it will find occurrences of either, whether or not they appear together.

Dragnet will search your entire disk or only a subset, and it can create one large file containing all successful matches. Also, it can run in background mode in *Microsoft Windows.* This feature is very handy, since text searches often take a long time.

If you have a large number of text files, both on- and offline, consider a catalog program. For example, *DISKCAT-5* archives files offline but maintains an online log. PearlSoft's *IN.SIGHT* keeps an index and catalog of all documents, whether on- or offline.

Compressing Data

Everything discussed thus far concerns finding files. The next concern is: how to fit all your files on your hard disk. By now you've probably removed everything that doesn't belong. But data, like work, expands to consume all available resources. You may use various methods to put more data on your hard disk before you run out of space. The two major forms of space-saving programs are:

- File combiners
- File compressors

File combiners. MS-DOS allocates file space in clusters. No matter how small a file, it uses a minimum of one cluster. Cluster size is 2K or more for hard disks. One way to save space is to combine a number of small files into one larger file. *ARC* and *XEQ* are two examples of file-combining programs.

XEQ.COM from Hardwood Software Associates is a run-library program available free from computer bulletin board systems or disk libraries. It adds program files to its 3K shell, then runs them from the shell. Each program takes up only 14 bytes more than its actual size. The maximum combined file size is 64K, but you can store multiple versions of the program, each with a different name.

Perhaps, for example, you have a small program file named TEST.EXE. You can move it into *XEQ*, then run it by typing **XEQ TEST** at the MS-DOS command prompt. Better still, create a batch file TEST.BAT containing only the single command above. When you type **TEST**, the program will appear to run just as it would normally. You don't even need to remember that you've moved it into *XEQ*.

ARC is a fancier file combiner and run-library. Unlike *XEQ*, it works on any kind of file. Also, it compresses as it combines. *ARC* is delivered as a self-unpacking library, as seen in Figure 5-6. It established a standard format for files moving to and from computer bulletin board systems (BBSs).

Often data files contain empty space—blanks, repeated characters or unused areas that can be removed from a file without loss. *ARC* checks which of four different compression methods yields the smallest output, then applies it. Output files are typically less than

Figure 5-6. Sophisticated Data Compression with Shareware Program *ARC*

```
E:\ARCHIVES>arc51
ARC51, version 1.07, created on 01/31/86 at 12:44:11
(C) COPYRIGHT 1986 by System Enhancement Associates; ALL RIGHTS RESERVED

This program delivers the latest version of ARC, along with its documentation.
Please wait while I unpack everything.

Extracting file: * unpacker *
Extracting file: ARC.DOC
Extracting file: ARC.EXE
Extracting file: ARC.TXT

E:\ARCHIVES>arc v pslist
Name          Length   Stowage   SF   Size now  Date       Time    CRC
============  ======== ========  ==== ========  =========  ======  ====
PRINT.TST         5632 Crunched  42%      3293   3 May 87   9:28p  A6AD
PSLIST.DOC        3111 Crunched  38%      1951  10 Jun 87   2:06a  5263
PSLIST.EXE       27514 Crunched  35%     18143  10 Jun 87   2:31a  F33D
        ====  ========           ==== ========
Total      3     36257            36%    23387

E:\ARCHIVES>
```

two-thirds the size of their originals. You must decompress and separate files before using them.

ARC also combines small programs as *XEQ* does, for compressing rarely-used data online, and for archiving old files offline. It is a shareware program from System Enhancements Associates. For $50, SEA will send you full documentation plus a good set of utilities. The commercial per-system license is $35.

For $20, you can get file compatability, faster performance, and more features. PKWARE offers a competing package, the *PKARC* compressor and the *PKXARC* extractor. As depicted in Figure 5-7, *PKARC* allows comments about files within an archive.

File compressors. A number of commercial file compressors are available. These work on one data file at a time, reducing its size as much as possible. The file must be decompressed before use. A file compressor that removes extraneous space from spreadsheet files is *SQZ!* from Turner Hall Publishing.

SQZ! compacts spreadsheets, including those for *1-2-3*, *VP-Planner*, *Q & A*, *Spreadsheet Analyst*, and *Reflex*. Most worksheets, by nature of their rectangular format, have many unused areas. Typically, *SQZ!* cuts eighty percent of their size.

Figure 5-7. Commenting a File Within an Archive

PKWARE's shareware program PKARC *is less expensive than* ARC *and offers features not found in* ARC, *such as file annotation.*

```
E:\ARCHIVES>pkarc vc pslist

PKARC   FAST!  Archive Create/Update Utility  Version 2.0   12-15-86
Copyright (c) 1986 PKWARE, Inc. All Rights Reserved.  PKARC/h for help

Searching Archive: PSLIST.ARC - ASCII/WordStar on LaserWriter

Filename       Comment                          Date      Time
--------       -------------------------------  ----      ----
PRINT.TST      WordStar 4.0 sample document     05-03-87  21:28:54
PSLIST.DOC     Feature summary, usage info      06-10-87  02:06:38
PSLIST.EXE     Text-to-PostScript translator    06-10-87  02:31:26
----           ------            ------  -----
0003            36257             23387   36%

E:\ARCHIVES>
```

The program is RAM resident; you simply pop up the *SQZ!* menu before writing a worksheet to disk. It is well-designed and coexists with many other RAM-resident programs, including *HAL* and *SideKick*. It sells for $80.

Other file compressors are general-purpose programs that work with any kind of file. Two examples are *Cubit* from SoftLogic Solutions and *Squish* from Sundog Software.

Squish version 1.3 is RAM-resident. It intercepts all disk activity, but only reacts to a compressed file. It collapses and expands disk data within a memory buffer. Compressed data goes between *Squish* and the disk, uncompressed data passes between *Squish* and the other software. Uncompressed files are unaffected and application programs are unaware that their files have been compressed. *Squish* costs $80.

Job Streams

Traditional floppy disk–based computer systems use two floppy disk drives, with each disk holding about 360,000 characters. On such systems, drive A usually contains program files and drive B contains data files. For example, drive A might hold your word processor; drive B, your document files.

Such systems are barely adequate to run one program at a

time. Your word processing files, for example, may actually fill up many disks. You organize them to keep related files together, on the same floppy disk, as much as possible. When this isn't possible, you swap data disks. Likewise, modern power applications typically require two to six disks. There's no possible way to have all features online at once. You must swap program disks as needed.

This bit of history points up a major advantage in hard disk systems: Hard disks keep many programs online at the same time. It's easy to combine different applications to yield a single, integrated output.

Suppose that your system is a PC AT with a hard disk. You started out with a wide-carriage dot-matrix printer running from the parallel port (LPT1). Later you expanded your system with a laser printer attached to your first serial port (COM1) and a mouse pointing device attached to the second serial port (COM2).

Once you have a laser printer, you use it. At least, as much as is reasonable. But your laser printer doesn't print on continuous paper. That's why you keep the dot-matrix printer around. So, you have installed your *1-2-3* speadsheet to send its output to the dot-matrix printer. Sometimes you want to print extemely wide speadsheets sideways, so you run a program that does just that.

Can you imagine the number of disks and disk swaps to perform this series of operations on floppy disks? On a hard disk, no swaps are needed; you simply move through the series of operations, one after another. This sequence of related programs is a *job stream.*

If you use the same program sequence frequently, or even occasionally, you can simplify the process by creating a batch file to control it. The batch file sets conditions for each program within the job stream, then restores the original conditions at the end. Figure 5-8 displays a batch file that

- Insures that the parallel printer is the active device
- Loads a mouse menu tailored for *1-2-3*
- Runs the Lotus software from the application's directory
- Suspends the mouse menu
- Runs the sideways-printing program
- Reactivates the serial laser printer

Figure 5-8. Using a Batch File to Turns a Sequence of Related Programs into a Unified Job Stream

```
C:\BATCH>type 1-2-3.bat
echo off
mode lpt1:
c:
cd c:\util\pop-ups
mousesys /2
m_lotus
c:
cd c:\lotus
lotus
mousesys /s
sideways
mode com1:9600,n,8,1,p
mode lpt1:=com1
exit

C:\BATCH>
```

Invoking that batch file by typing 1-2-3 is simpler than remembering to run all those commands. It also insures that important steps, like reactivating the laser printer for subsequent word processing, are not forgotten.

The batch file in Figure 5-8 uses no parameters. You must give each program the filenames and internal commands it needs to produce the desired results. For instance, if you don't want to print anything this time, you merely exit the sideways-printing tool.

Unifying Your Environment

Once you realize how many different programs you can run from your hard disk, you can begin to appreciate the need for a common interface. Interface here means the way you interact with the computer. Without a concerted effort, the steps to run each program are different. You must remember which directories to change to, which PATH and APPEND commands to set up, and how to give instructions to each program.

Batch files are a first step at standardization. Rather than remembering PATH, APPEND, and other information, you need only remember a batch file's name. However, much more unification is possible. In this section three approaches to unification will

be discussed:

- Menu programs
- Macro processors
- Operating environments

Menus. Menus present a friendly face to users. Rather than a forbidding *C:*\> , the operator sees a somewhat helpful screen, such as in Figure 5-9.

Figure 5-9. Helpful Menu Directions

A menu can simply tell you what command to type.

```
                         K.I.S.S. MAIN  MENU

[STOCK   ] *  Inventory control:  Material, Issues, Builds, Counts

[BILLING ] *  Accounts receivable:  Customers, Invoices, Payments, Dunning

[CHECKS  ] *  Accounts payable:  Vendors, P.O.'s, Vouchers, Reports

[PAYROLL ] *  Payroll accounting:  Masters, Time cards, Up-loading

[WORDS   ] *  Word processing:  Editing, Proofing, Printing, Forms design

[CALCS   ] *  Financial planning:  Modeling, Estimating, Spreadsheets

[PHONE   ] *  Communications:  TELEX, Electronic mail, Corporate Network

[FILES   ] *  Information:  Forms definition, Data entry, Query, Reports

[SYSTEM  ] *  System maintenance:  Media and Hardware Setup

 Type the [COMMAND] between the braces ("[]"), then press the |Enter| key.
```

Menus are no panacea; there are conditions under which they can be frustrating. If you're the only person using your system, and you use the same small set of programs over and over, menus can become just one more layer getting in the way. Furthermore, it takes time and effort to set up a thorough, useful set of menus.

However, if you share your system with others, or if you use your system only occasionally and want to get into and out of it with minimal review and manual reading, menus can provide much help.

Figure 5-9 shows the simplest kind of menu. The menu text is stored in a file and is displayed by a TYPE command in the menu's batch file (see Figure 5-10). Such a menu merely offers information. The operator must manually enter commands to start all applications. But even a simple, read-only menu is a vast improvement over no menu at all.

Figure 5-10. Menu System Run by a Minimal Batch File

```
C:\BATCH>type kiss.bat
echo off
REM K.I.S.S. MAIN MENU
cls
type c:\batch\main.kis


C:\BATCH>
```

From such simple beginnings, many menu improvements are possible. You can build fairly complex menus using batch files or macros. Appendix C includes a complete nested menu system, featuring:

- Nested menus: Menu selections lead to other, more detailed menus or to program loading. Finishing a nested menu returns to the previous menu.
- Selection and program loading: Press a key to indicate your choice, and the menu program loads the selected software or a batch file which in turn loads the software.
- Parameter passing: The program not only starts your program running, but also passes it file or command information.

The menus are already set up for some major software packages. You can easily tailor them to your specific requirements.

Further possible enhancements in self-developed menus include:

- Password protection: Users must know a password before they can use a program or file.
- Online help screens.
- Aesthetic improvements: Attractive multicolor screens, boxes, shadows, reverse video, highlighting, graphics.

If do-it-yourself programs aren't enough, a number of commercial and public domain programs will help you create custom menus. Typically these programs guide you through the menu-creation process, prompting you for needed information. Some are very basic. Many include sophisticated file, directory, or screen management features. A few offer advanced protective features, such as restricted access to directories or files, and the inability to reach the DOS command.

For now, our focus is on the simplicity created by the presence of a common menu system throughout your system. Here is a brief look at several menu programs, in order of increasing complexity.

AUTOMENU is a rudimentary shareware menuing program. It lets you create attractive pull-down menus with minimal fuss. Use your word processor to create an ASCII text file containing menu information. AUTOMENU does the rest. Menu choices can call programs, batch files, or other menus.

Program Director, from Power Up! Software, is easy to install and run. You can set up menus for a single user or multiuser menus with three levels of passwords. You can continue to use your AUTOEXEC.BAT, simply add a call to the *Program Director.* You can call batch files, or you can set up what are essentially batch files within the program. If you use the latter method, the program automatically reloads after each application program. You can create online help. You cannot do any kind of sophisticated parameter passing.

Direct Access, from Delta Technology, includes more features. The main menu can have up to 20 choices, with ten-selection submenus available for each. It also allows three levels of passwords. One unusual feature is an audit trail which can track your time on different projects or track how different people use the system. Selections can call programs, batch files, or BASIC programs. You can also build custom batch files within the program. You can prompt users to insert a disk or to type in parameters. There is no built-in DOS menu, but you can easily create one.

Flash-Up Windows, from the Software Bottling Company, takes a pop-up approach to menu design. The idea is to put pop-up menus inside all your application packages, providing a unified, easy-to-use interface. The program also includes online, context-sensitive help and ties into a macro processor for a powerful, integrated package.

One of the most complex menuing programs is *Hot!*, distributed by Executive Systems. The program features:

- Password protection
- Multiple nesting levels for menus
- Pop-up menus available at any time, including menus attached to function keys
- Passing of keystrokes to programs after calling

The program also includes a number of nonmenu features such as desktop utilities and file management features, but these are outside our current focus.

One unusual and highly valuable feature is automatic menu creation. On request, *Hot!* scans your disk for program files and creates menus and submenus calling these programs.

Every menu option can have its own pop-up help line. Menu selection is accomplished by moving the cursor to the desired choice and then pressing Enter. Unfortunately, the manual, online tutorial, and online help are all less than ideal. It's hard to figure out exactly how to use the program's power. For example, a special editor, the Chef, modifies menus on the fly, but it's hard to learn and tedious to use. Likewise, the automatic menu creation system does not include adequate error protection. One slip of the finger can destroy an entire series of menus.

Macro processors. How do you give instructions to the various software programs you run? Some programs use menus, some use control sequences, some use function keys. How do you remember to type *E* to *exit WordStar* and *Q* to *quit DBase*?

What can you do to avoid repeatedly issuing the the same commands within a program? Batch files can move you from program to program for frequently repeated job streams. Using redirection, you can pass keystrokes to programs. When you redirect the keyboard input, you cannot do any actual data entry. Redirection precludes variability. It is suitable only if the program always does the same thing.

Some programs offer internal macro capability. Internal macros define a single key to represent a series of keystrokes, such as a program's internal command sequences. However, there are problems with internal macros:

- Many programs don't offer any internal macro capabilities.
- The macro capabilities in many programs are hard to use. In *1-2-*

3, for example, you must memorize the complete command sequence and reproduce it exactly. *WordPerfect* lets you capture command sequences as you perform them, but it offers no way to edit sequences later. (Both these applications offer extra-cost add-on modules to remedy these shortcomings).
- Even in the best programs, macro processing differs greatly from program to program. If you want sophisticated usage from a number of programs, you must learn a wide variety of macro formats, commands, and editing modes.
- Internal macro processors work only within a single program. Ideally, you'd like to combine MS-DOS commands with internal macros, possibly for several different programs, all within a single command sequence.

Fortunately, there is a solution to all these problems and incompatibilities. Macro generators are stand-alone utilities, which means that they're separate from your application programs. They're memory resident, which means they stay in memory, in the background, through all the comings and goings of your application programs. They let you call up any predefined sequence of keyboard input with a single name or key.

Typically macro generators are part of larger utility programs. These fall loosely into two categories:

- Keyboard enhancers
- Desktop organizers

Don't be concerned if these terms are unfamiliar. They'll be examined in more detail later. Features in the best macro programs include:

- Learn mode: You tell the program to record your keystrokes as you enter them, then save the sequence under an assigned name or key. The very best macro processors even provide for after-the-fact learn mode. This means that the program records all keystrokes, even if you didn't specifically request it. If you decide, after entering a sequence, that you'd like to save it as a macro, you simply assign a name or key and you're in business.
- Edit mode: You should be able to change macros or copy them and then use the copies as the basis for new macros.
- Nesting: Macros should be able to call other macros, programs, and batch files.

- User response: This lets you prompt for user input, wait for a response, then incorporate the response for further processing. Often you can assign default response values. User input is invaluable for increasing the power and flexibility of macros.
- Context sensitivity: In context-sensitive macro processors, the meaning of a macro call depends on where you are when you call it. Typing the letter *P* might call printer macros in both *1-2-3* and *DBase*, but the actual commands sequences would be different in each program. Very few programs have this level of sensitivity. However, some programs can distinguish between DOS and program modes.
- Abort and interrupt mode: You should be able to stop a macro in mid-stream or interrupt it temporarily.
- Macro menus: This is a formatted list of all the macros you've created. If you can't remember the name of a macro, it's helpful to flash up a menu, then pick the desired macro from the list.
- Menu commands: Some macro processors include commands to facilitate building your own menus. Typically, the menu macros then call additional macros to execute the various menu choices.
- Loops and pauses: Good macro languages include all the looping (FOR, REPEAT UNTIL, and DO WHILE) and control features of any modern programming language. You should be able to pause a specified amount of time.
- Single-stepping: When you first create a complex macro, it's likely to have errors. Single-stepping lets you walk through the macro one step at a time to quickly find and correct your mistakes.
- Protection: You can't change or remove a protected macro unless you have the proper password to remove the protection.

Keyboard enhancers. As mentioned above, macro processors typically come as part of larger utility packages. Their most common home is in a family of programs called keyboard enhancers, which redefine keyboard input for greater ease and compatibility.

For example, suppose you use two different word processors because you need features from both. In program one, you type **Esc, S** to issue a search command; in program two, you press the F2 function key. Wouldn't it be easier and less confusing to use F2 in both programs? A keyboard enhancer lets you redefine the F2 key. When you press F2 in word processor number one, the keyboard enhancer translates it and sends the *Esc, S* sequence instead.

ProKey from Borland International was the first commercial keyboard enhancer, and it was an overnight success. A later offering is *SuperKey*. *Superkey* retails for $100, includes all *ProKey* features plus pull-down menus, file encryption, and a DOS command editor. Both *ProKey* and *SuperKey* feature excellent macro capabilities.

Other keyboard programs with strong macro processors include *KeyWorks* from Alpha Software and *SmartKey II Plus* from Software Research Technology.

Desktop organizers. Another source of macro capabilities is a class of programs known as desktop organizers. These products mimic the typical business desktop: They include a calculator, calendar, clock, notepad, and telephone directory. Only the most powerful desktop organizers include macro processors: Lotus's *Metro*, WordPerfect's *WordPerfect Library*, and TEK Microsystem's *HQ*. All are at least partially memory-resident, so you can call them into use with a single keystroke, as in Figure 5-11.

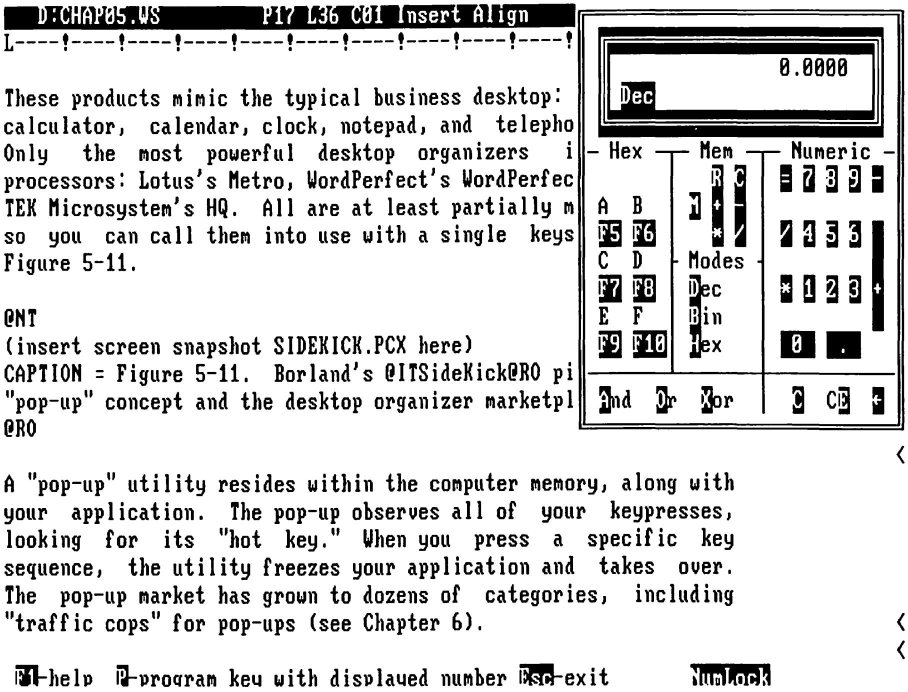
A pop-up utility resides within the computer memory, along with your application. The utility observes all your keypresses, waiting for its *hot key*. When you press a specific key sequence, the utility freezes your application and takes over. The pop-up utility market has grown to dozens of categories, including traffic cops to direct the flow of pop-up utilities (see Chapter 6).

Operating environments. Macro processors are a big step up from simple batch files or menus. An even bigger step up is to full-scale operating environments or desktop metaphors. The environment concept is to offer a single unified way of interacting with the computer. In theory, environments offer:

- A common *look-and-feel* for all programs
- Multitasking: The ability to run more than one program at a time
- Cut-and-paste: The ability to transfer data between programs
- The ability to switch between programs
- Windowing: Screen management enabling applications to share the console screen under user control

Not all operating environments provide all these features in their fullest forms.

Figure 5-11. Pop-Up Concept and Desktop Organizer Marketplace, as Pioneered by Borland's *SideKick*



The ancestor of all microcomputer operating environments was not an IBM PC- compatible product at all. Apple Computers offered the first commercially available desktop metaphor for microcomputers on the Macintosh and its predecessor, the Lisa. These in turn were largely inspired by the Xerox Star system developed at the Palo Alto Research Center (PARC).

The Star system displayed most of the features of current operating environments. Most noticeable was its consistent user interface, screen icons, graphics symbols which represent processing options, and user selections made with a mouse (an input device which moves a screen pointer in response to physical movement across a desktop or other flat surface). Mice normally have from one to three buttons, with two being the standard for the PC world.

Mouse operation consists of four actions:

- Click: Pressing a mouse button while the cursor is positioned on an icon or menu line.
- Double-click: Two clicks in rapid succession on a single icon or menu line.
- Shift-click: A click while holding down the shift key. Typically this is how you select several icons at once: You hold down the shift key, then move to each desired icon in turn and click on it.
- Drag: Moving to an icon, pushing down the mouse button, then moving to another location while holding down the button. When you release the mouse button, the icon becomes fixed in its new location.

In the Star/Lisa/Macintosh environments, and now in the various environments available for the IBM and compatibles, the user moves the screen pointer to the desired icon, then clicks to indicate the selection. More complicated choices appear on pull-down menus. Main menus list choices along the top of the screen. By moving the screen pointer to a menu and clicking on it, the user can expand, or pull down, the menu to see all available choices. Any options that aren't allowed are *grayed,* as is the first item of the GEM Desktop screen in Figure 5-12. Another click on the desired option completes the selection.

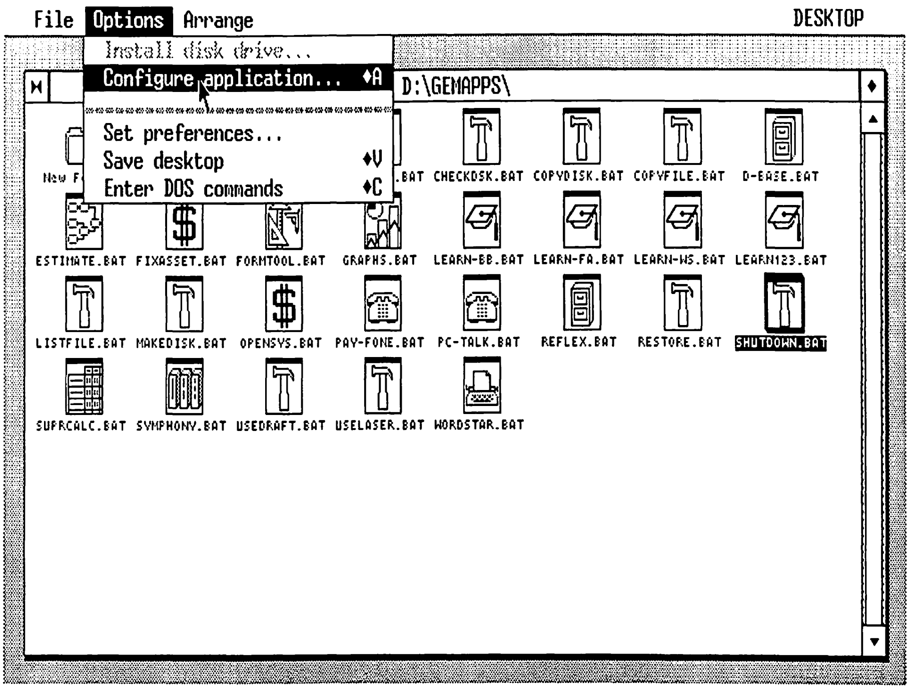
In most of these desktop environments, programs present their information within regions of the screen, called windows, which users can expand, contract, move, or hide. Multiple windows can exist on the same screen. Users change applications or files by simply moving the mouse pointer to the application window and clicking.

The visual menu system is easy to learn. Computer novices can perform useful work in their first hour on the machine. On the Macintosh at least, nearly all programs communicate with the user through the same interface to ease the transition to new applications.

With this background in mind, here is a look at the four major operating environments in the IBM PC-compatible world:

- GEM Desktop
- MicroSoft *Windows*
- TopView
- DesqView

Figure 5-12. Example of a Grayed, Unavailable Option

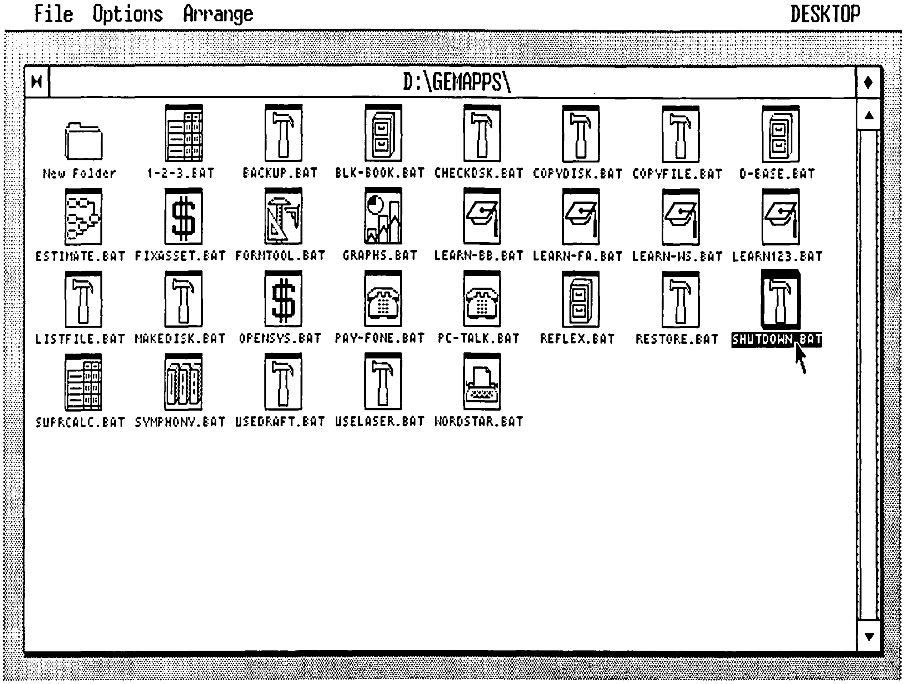


GEM. Digital Research's Graphic Environment Manager (GEM) closely resembles the Star and Macintosh environments. Indeed, Apple prevailed upon Digital Research in 1986 to change certain parts of GEM's visual and procedural aspects to avoid infringing upon the look and feel of the Macintosh interface. As you can see in Figure 5-13, GEM provides a far different look and feel than the traditional text-only MS-DOS command line. It requires a graphics video card, such as the IBM CGA or EGA, or a Hercules HGC.

Surprisingly little effort is needed to move an operation cleanly from MS-DOS to GEM. All but one of the icons within the desktop of Figure 5-13 are batch files. They create the same job streams we discussed both here and in Chapter 4. All we do to turn an MS-DOS batch file into a useable GEM applicationn is:

- Copy it into the \GEMAPPS (or other designated) directory
- Select it with a single-click, as in Figure 5-13
- Double-click the *Configure application* pull-down menu (seen in Figure 5- 12)

Figure 5-13. MS-DOS Batch-File Job Streams Presented as Icons by GEM

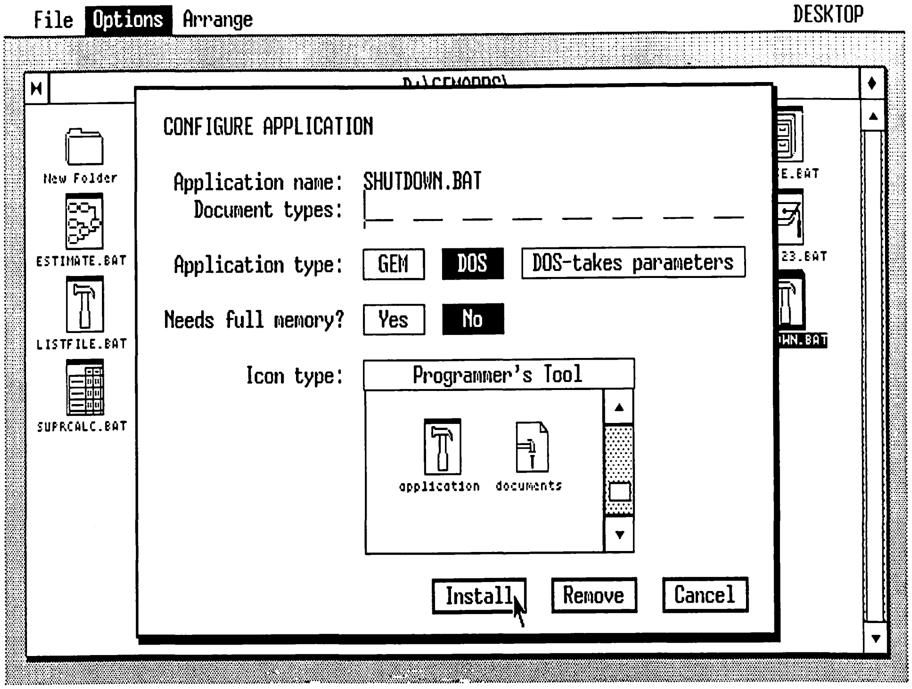


- Indicate any default document types (predefined filename extensions such as .WKS or .CAL for spreadsheets, .DBF or .NDX for databases), if it needs full memory (*Yes* is always a safe answer and only costs a slight slow-down during program loading), select the icon to be associated with the batch file from a list of predefined icons, and click on *Install* as in Figure 5-14
- Save the Desktop information (the fourth item in Figure 5-12)

Although designed to work with a mouse, GEM can operate from the keyboard as well. The keyboard cursor arrows are used for mouse movements and the Home and End keys mimic click and drag mouse operations. Keyboard shortcuts appear on the right side of pull-down menus. Looking back at Figure 5-12, you can see that pressing Alt-A would also install the selected application.

Double-clicking on a program or batch file starts it running. You can also start an application program by double-clicking on

Figure 5-14. The GEM Desktop needs little information to install an application.



one of its data files, but this only works if the program uses consistent filename extensions, which you specified as the default *Document types* during installation (see Figure 5-14).

Most file- and directory-management functions are easy to do within GEM. You can change directories, copy single or multiple files, or whole disks, change filenames, and format disks. Other MS-DOS functions are available in a special DOS window.

Since GEM is a graphics application, many text-only desktop utilities cannot work with it. GEM provides most of the common desktop functions, including a calculator, clock, and print spooler (copies print files to disk for subsequent printing). Other than printing in the background while you run other programs, GEM does not provide multitasking.

Windows. MicroSoft *Windows* is another Star-like operating environment. It therefore requires a graphics display, such as an IBM CGA or EGA, or a Hercules HGC. It runs best with a mouse,

using clicks, double clicks, shift-clicks, and drags similar to GEM, but it, too, can be controlled from the keyboard.

Windows is the product Microsoft has been proclaiming as essential in all versions of MS-DOS after 3.3. Indeed, the Presentation Manager of OS/2 is version 2.0 of *Windows* with somewhat different internal commands.

It simulates multitasking by allowing you to open more than one window at a time. You can also expand, shrink, and move windows. If you want to move a window out of the way temporarily without stopping the application, you drag it to the bottom of the screen, and it changes from a window to a small icon. When you finish with an application, you simply close the window it is running in. The window or icon disppears from the screen, freeing memory for the next application.

The program includes:

- A shell program, the *MS-DOS Executive*, to handle formatting disks, making, changing, and sorting subdirectories, printing, copying, renaming, and removing files
- Desktop utilities: calculator, notepad, clock, and clipboard
- Microsoft *WRITE* word processing program
- *PAINT*, a graphics package

The clipboard is a cut-and-paste tool. Use it to copy text or data onto the clipboard, then copy it into another data file.

Like GEM, you either click on a program icon to run it, or select it with the keyboard. You can also click on a data file if it belongs to a known *Windows* program. If so, the program starts running, using the selected file for data.

There is no theoretical limit to the number of windows or applications open at any time. Since PCs have a single processor, only a single task runs at a time. Rapid movement between tasks (*context-switching*) gives the appearance of multiple tasks running concurrently.

Windows knows which tasks are in suspended animation waiting for input from the user or other tasks. Inactive applications are swapped out to data memory, or to disk if memory is insufficient to keep all windows resident. When you reactivate a window, it starts processing from where it left off. Application swapping can be very

time consuming, even with a fast processor and hard disk, so you should have a great deal of memory to operate in the *Windows* environment.

Versions of *Windows* prior to 2.0 used *tiled* windows, where nothing overlapped. Any change in a window size or position caused a redraw of the entire screen. Because all tasks were visible and many could be running, the screen was also constantly being updated. With *Windows* 2.0, Micrsoft began overlapping windows to gain speed by reducing the need to redraw.

Installing an application into *Windows* requires much more information than installing in GEM. Since *Windows* does more, it must know more. This needed information resides in a Program Information File, or PIF, similar to those used by TopView (below). Microsoft provides several dozen PIF's with *Windows*. However, PIFs are no panacea.

Programs that write directly to the video hardware for speed are incompatible with split-screen windows. These programs, which include *1-2-3*, *WordStar*, and many others, insist on taking over the entire screen. This is not a problem with the multiple *virtual* machines available within the 80386-based version of *Windows*.

TopView. *TopView* was the official IBM environment manager until the announcement of the OS/2 Presentation Manager (a close cousin to version 2.0 *Windows*). This orphan is often clumsy and heavy-handed.

TopView supports mouse or keyboard operation. A text-only environment, it runs on almost any kind of display, whether color or monochrome, graphics or not. It relies on menus and submenus. You must make selections by moving the cursor to the desired menu choice—there are no keyboard shortcuts.

In principle, *TopView* supports simulated multitasking, windowing, and cut-and-paste transfer of data between programs. In reality, all these features have substantial limitations. Many programs require major modifications to work properly with *TopView*. Three levels of compatibility exist:

- Fully compatible programs have been written or modified to conform fully to *TopView*. The number of these programs is growing, but very slowly.
- IBM compatible programs that have not been specifically tailored to *TopView*. However, they must use standard DOS input/output routines and generally conform to DOS conventions.

- Nonstandard programs that use nonstandard disk or screen processing or are otherwise incompatible with DOS standards. Unfortunately, many popular applications, including *WordStar* and *1-2-3,* fall in this category.

Nonstandard programs work under *TopView* to some extent. They cannot use partial-screen windows: Their screen displays cover the entire screen. They cannot multitask except as the main program. If you switch to another program, they stop operations rather than continuing to run in the background.

Cut-and-paste is incomplete. You can cut, or copy, data from almost any application. However, you can only paste, or import, data into fully compatible *TopView* programs. As there are few of these, and as there are not likely to be many more, the cut-and-paste feature is largely useless.

Each application requires a Program Information File (PIF) to introduce it to *TopView.* The PIF lists the startup program name, memory requirements, swapping and foreground/background limitations. It includes technical information about screen pages, interrupt vectors, window sizes, and more.

A few IBM program PIFs are included with *TopView.* Some applications now come with preestablished PIF files. For most applications, however, you must create the PIF yourself. As you can see from the above paragraph, much of the information necessary to create the PIF is not easy to come by.

IBM supplies a long listing of PIF data for various application programs. However, you must enter the information into the PIF editor program. And if your application program isn't on the list, you must simply make your best guess at the proper PIF entries. This seems ludicrous in an environment designed, among other things, to simplify computing for novice users.

DesqView. DesqView shows what *TopView* could and should have been. It is similar to *TopView* in many ways:

- They're both menu based.
- They both run on graphic or nongraphic display consoles, in either color or monochrome.
- They both use standard keyboard input, although a mouse is also an acceptable input option with *DeskView.*
- It uses *DesqView* Information Files (DVPs) to store program information, much like *TopView*'s PIFs.

• They both support windows, multitasking, memory, directory and file management functions.

DesqView's documentation, ease of installation, and ease of use outstrip *TopView*. For example, the *DesqView* installation procedure asks only two questions. Once you identify your monitor and mouse, other installation information can wait.

Another interesting feature is *DesqView*'s automatic identification procedure. The install program searches your hard disk for applications it recognizes. For these, it automatically creates DVP files with correct path information. You need to double-check its accuracy, but it's easy to make changes if needed. *DesqView* also uses the information in any *TopView* PIF files it finds to create its own DVP files.

You can easily alter the contents of DVP files. For instance, many people assign the full screen to their major application programs, such as word processors, database managers, and spreadsheets. Pressing the Alt key engages the *DesqView* menu, which then allows resizing or switching windows.

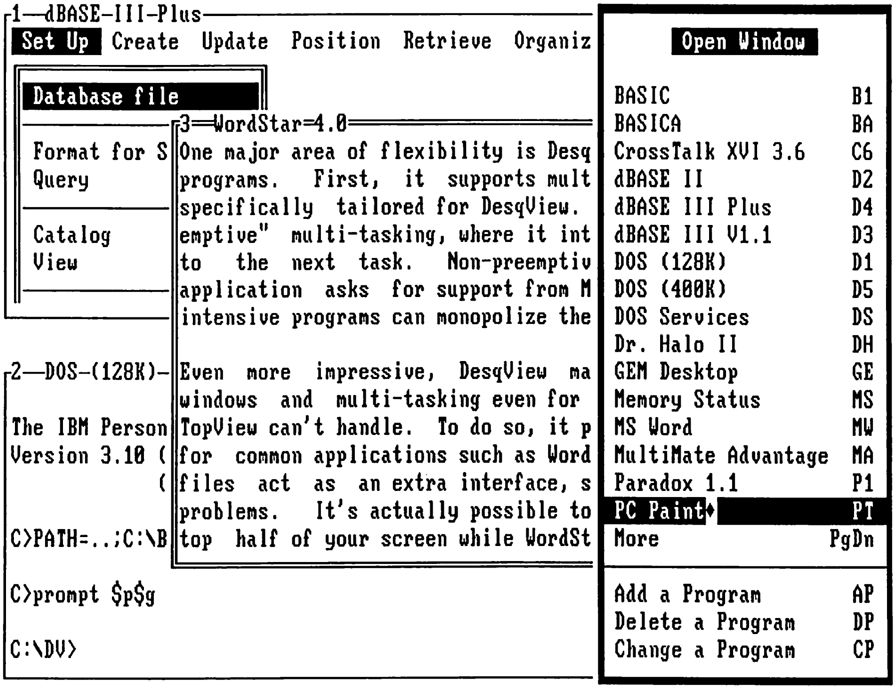
Unlike *TopView*, *DesqView* always lets you enter short keyboard commands as an option to highlighting menu choices. As shown in Figure 5-15, the keyboard shortcuts appear along the right edge of the menu.

One major area of flexibility is *DesqView's* handling of difficult programs. First, it supports multitasking of programs not specifically tailored for *DesqView*. *DesqView* offers preemptive multitasking, where it interrupts to pass the baton to the next task. Nonpreemptive systems wait until an application asks for support from MS-DOS, so certain programs can monopolize the system.

Even more impressive, *DesqView* manages to support multiple windows and multitasking even for many nonstandard programs *TopView* can't handle. To do so, it provides special *load files* for common applications such as *WordStar* and *1-2-3*. These load files act as an extra interface, smoothing disk and console problems. It's actually possible to watch *1-2-3* running on the top half of your screen while *WordStar* runs on the bottom half. On some PCs they both run more slowly that you would expect, but the ability to run both is impressive nonetheless.

DesqView performs cut-and-paste better than *TopView*. You can transfer data between almost any two files. Furthermore, *DesqView*

Figure 5-15. *DesqView* 2.1 provides multitasking support for both text and graphics programs, working via mouse selection or keyboard commands.



takes the additional step of helping you reformat data where appropriate. After all, different word processors, databases, and spreadsheets keep data in different formats. Transferring incompatible data is nearly useless.

Coupled with QuarterDeck's extended memory manager (*QEMM*), *DesqView* 2.0 can provide multiple virtual 8086 PCs within a single 80386-based computer. *QEMM* allocates regular AT-type extended memory (no need for the expensive *EMS 4.0* or *EEMS* RAM) using the 80386 virtual-addressing mode. Depending upon the type of video adaptor and what resident software you run, you can get nearly 500K of free space for MS-DOS applications, which themselves can use EMS or EEMS memory. Your only practical limiting factors are the total memory requirement (about 512K per virtual PC) and the overall system slowdown. When you share a single CPU chip among five to seven virtual computers, even a 20Mhz 80386 seems sluggish.

DesqView provides an easy, readable manual with an excellent tutorial. It offers macro capabilities; extensive file, directory, and disk management support; and many other pluses. With the advent of EMS 4.0 memory drivers, *DesqView* 2.1 will run up to 32Mb of programs even on an 8088-based IBM PC.

Managing Multiple Systems

So far this book has dealt with single computer systems. The last topic in this chapter is multiple computer systems. Whether you work alone or with others, at home or in an office, you may well find yourself managing more than one system at some point.

How is it done? There are two basic choices:

- Optimize each system for the exact demands put upon it, without regard to standardization between systems.
- Treat all machines as standard, interchangeable sites.

Most experts opt for the second alternative, known as the *generic or virtual software sites*, which requires the same tools and interface on all systems. This involves:

- Menu or shell programs
- Support utilities for managing files, directories, keyboard, and screen
- Backup, restore, and recovery programs
- Batch files
- Macros
- Software or hardware protection
- Applications packages

If you redefine keys on one system, do so on all. If you put menus on one system, do so on all. In general, however you use tools on one system, do so on all. There may be some slight loss in efficiency or disk space. However, the advantages far outweigh the disadvantages:

- Each system becomes a generic software support site, usable by anyone at any time without retraining.
- Anyone can work on any system, simply by copying data files.
- Standardized procedures reduce mistakes.
- Minimal switchover time if one system needs repair.
- Problems are easier to troubleshoot because of a common base of experience and knowledge.

Conclusion

This chapter has introduced a great deal of information about day-to-day management of hard disks. The focus has been on ease of use through simplicity and unified ways of doing things.

The next chapter focuses on speed: making it faster rather than making it easier. Clearly the two overlap. If it's simpler, cleaner, and easier, it's going to be faster in the long run.

To help you get organized and stay organized, use these checklists to evaluate organization, macro processors, and operating environments.

Prepared by ______________________________

Date ______________________________

System(s) ______________________________

Who is responsible for the continued organization of this disk?

____

Controlling Clutter

	Yes	No
Do we have consistent file-naming conventions?	____	____
Are these written down?		
Where: ______________		
Do we keep directories small (20 to 60 entries)?	____	____
Do we regularly remove old files, on-line backups, or other data which doesn't belong on the hard disk?	____	____
Do we have a \SCRATCH directory for temporary files?	____	____
Do we keep all appropriate data on floppy disks?	____	____

Tracking and Identifying Files

Do we/should we use any of the following to help identify and keep track of files:

	We should		*We do*		*Program*
	Yes	*No*	*Yes*	*No*	*name*
Improved directory list (*SD.COM*)	____	____	____	____	________
Expanded filename utility (*PCEasy II*)	____	____	____	____	________
Date-oriented file tracker (*Dayflo Tracker*)	____	____	____	____	________
Paste-on notes (*SmartNotes*)	____	____	____	____	________
Universal context-sensitive help (*Rescue*)	____	____	____	____	________
Keyword file trackers (*Zoo Keeper*)	____	____	____	____	________
Text searchers (*Dragnet, Norton Util. TS*)	____	____	____	____	________

Combining and Compressing Files

Do we/should we use any of the following to combine or compress files:	*We should* Yes	No	*We do* Yes	No	*Program name*
Program to combine small programs (*XEQ*)	____	____	____	____	________
Data file combine/compress programs (*ARC*)	____	____	____	____	________
Generalized file compression (*Cubit, Squish*)	____	____	____	____	________
Specialized file compression programs (*SQZ!*)	____	____	____	____	________

Overall Consistency

Do we/should we use any of the following to provide a consistent user interface:	*We should* Yes	No	*We do* Yes	No	*Program name*
Menu programs, homemade or commercial	____	____	____	____	________
Macro processors	____	____	____	____	________
Environment managers	____	____	____	____	________

Standardization Among Systems

	Yes	*No*
Have we organized all our systems for maximum standardization, so users can easily move from one software support site to another?	____	____

Program name ____________________

Person doing the analysis ____________ Date: ____________

Macro Features

Does this program offer:	Yes	No
Learn mode (learns by recording your keystrokes)	____	____
After-the-fact learn mode	____	____
Edit mode (you can copy and change macros)	____	____
Nesting (macros can call other macros)	____	____
Interrupt mode (you can interrupt a macro and resume later)	____	____
Ability to abort macros in process	____	____
Single stepping (slow operation to test functions)	____	____
Macro menus (lists of available macros)	____	____
Loop control (FOR, WHILE, and REPEAT commands)	____	____
User input, with WAIT and PAUSE	____	____
Menu commands available as part of the macro language	____	____
Ability to protect macros	____	____

Other valuable features:

____

____

____

Larger Program Features:

Does this program offer:	*Yes*	*No*
Windows	____	____
Online help	____	____
Menus	____	____
Desktop processing	____	____
Calendar	____	____
Clock	____	____
Appointment scheduler	____	____
Notepad	____	____
Database manager	____	____
Phone lists	____	____
File management	____	____
Directory management	____	____
DOS shell	____	____
Keyboard control	____	____
Screen dimmer	____	____
Other valuable features:	____	____

____

____

____

Environment name ______________________________________

Person doing the analysis ______________ Date: ______________________

Environment Features

	Yes	*No*
Automatic program recognition/fast start mode	____	____
Concurrency	____	____
Macro processing	____	____
Workable cut-and-paste	____	____
Data reformatting during cut-and-paste	____	____
Program startup by selecting data files	____	____
Desktop functions		
Clock	____	____
Calendar	____	____
Notepad	____	____
Phone list	____	____
Print spooler	____	____
Other:	____	____
Software included:		
Word processing	____	____
Graphics	____	____
Database manager	____	____
Other:	____	____

Quality

	Poor	*Fair*	*Good*	*Excellent*
Rate this program on:	___	___	___	___
Memory requirements	___	___	___	___
Speed	___	___	___	___
Price	___	___	___	___
Ease of installation	___	___	___	___
Ease of learning	___	___	___	___
Ease of use	___	___	___	___
Documentation	___	___	___	___
Online tutorial	___	___	___	___
File management	___	___	___	___
Directory management	___	___	___	___
Handling "difficult" programs	___	___	___	___

Compatibility

	Poor	*Fair*	*Good*	*Excellent*
How compatible is this program with our:	___	___	___	___
Console display (graphics/nongraphics)	___	___	___	___
Input device (mouse or keyboard)	___	___	___	___
Word processors	___	___	___	___
Database managers	___	___	___	___
Spreadsheets	___	___	___	___
Graphics managers	___	___	___	___
Project managers	___	___	___	___
Accounting programs	___	___	___	___
Utilities	___	___	___	___
Other:______________________________				

Chapter 6

Performance Boosts

Performance Boosts

This chapter should be fun. After all the laborious work of the preceding chapters, you finally get to play a bit. You'll see how to soup up your hard disk system.

There are many ways to speed up your system using hardware, software, and sheer cleverness. After that, there's a quick look at the variety of utilities available to streamline your hard disk usage. The overall result of all these tools and tricks is a faster, more satisfying system—a hotrod instead of a jalopy.

You can boost performance using a variety of methods:

- Hardware speedups
    - RLL disk controllers
    - Faster processing
    - Expanded memory
- Speed access with ramdisks and disk caches
- Optimize the interleave on your system
- Reorganize your disk regularly for maximum efficiency
- Upgrade system tasks with good tools
    - File finders/path extenders
    - MS-DOS command editors
    - File managers/MS-DOS shells
    - Keyboard/screen controllers
    - RAM-resident software managers
- Maximize hard-disk possibilities with specialized tools
    - Translators
    - Print spoolers
    - Task-switchers

Are Hardware Upgrades Worth the Price?

Your computer is a complex system integrating one or more central processors with memory, display console, keyboard, data storage devices, transfer busses, and more. Overall system speed depends

on many interrelated hardware factors:

- Rated processor speed
- Actual processor speed as controlled by the clock crystal
- Available memory
- Memory speed
- Data transfer width
- Disk data transfer rate

In some cases, you can significantly increase throughput by speeding up one or more components. However, this is not an open-ended process. Sooner or later, you will hit your system's limits. Data transfer width in particular is outside your control. Most computers are equipped with RAM having access times adequate for their current processor speed, and not quick enough for anything faster. Therefore, it doesn't make sense to make your system faster than your memory and transfer width can support.

Much of this chapter aims to help you figure out what does make sense in the hardware speed-up arena. Topics covered include:

- Faster disk controllers
- Processor and memory speedups
- Increased random access memory (RAM) storage space

Since so many performance *tweaks* are trade-offs, you need to measure their overall impact on your system. Performance measurement software helps you evaluate your choices.

Performance Measurement: The Big Picture

Consider performance measurement as part of your performance strategy. Before you can know how far you've come, you must know where you started. Various programs measure the performance of personal computers. Comparing dissimilar computers with the same benchmark can be misleading. An 80386-based computer will offer capabilites that cannot be measured by software that evaluates 80286 systems. But running the same test on a single computer both before and after a tweak can provide valuable information.

Figure 6-1. *Norton*'s SI (SysInfo) measures performance in terms of a PC XT.

```
SI-System Information, Version 4.00, (C) Copr 1984-87, Peter Norton

        Computer Name:  IBM/PC-AT
     Operating System:  DOS 3.20
  Built-in BIOS dated:  Friday, June 20, 1986
       Main Processor:  Intel 80286          Serial Ports:  2
         Co-Processor:  None               Parallel Ports:  1
Video Display Adapter:  Monochrome
   Current Video Mode:  Text, 80 x 25 Monochrome
Available Disk Drives:  6, A: - F:

DOS reports 640 K-bytes of memory:
   288 K-bytes used by DOS and resident programs
   352 K-bytes available for application programs
A search for active memory finds:
   640 K-bytes main memory     (at hex 0000-A000)
    51 K-bytes display memory  (at hex B000-BCC0)
    11 K-bytes display memory  (at hex BD40-C000)

  Computing Index (CI), relative to IBM/XT: 11.2
       Disk Index (DI), relative to IBM/XT: 2.5

Performance Index (PI), relative to IBM/XT: 8.3
```

Norton Utilities' SI (System Information, or SysInfo for short) ratings are often quoted in computer advertisements. Figure 6-1 shows the SI results of a 10MHz PC AT clone system with fast enough RAM to run with no wait states. Its Computing Index (CI) is a respectable 11.2 times that of a stock IBM PC XT. The 20Mb hard disk gets Disk Index (DI) of 2.5 because the drive has an average track-to-track seek time of 40ms rather than the 85ms of an XT. The overall SI rating is 8.3; this machine is fast.

Figure 6-2 is the SysInfo for the same system, but after the hard disk sector interleave had been changed from the factory-standard 1:5 to the optimum 1:2. (How to determine your optimum interleave will be discussed later in this chapter). Notice that the Disk Index jumps from 2.5 to 3.2 because of faster sequential sector access. An already-fast machine boosted its SI rating to 8.5.

The SI program uses a proprietary algorithm to derive its performance indices. You do not know precisely what the Computing

Figure 6-2. As measured by *Norton*'s SI (SysInfo) program, going from a 5:1 to a 2:1 hard disk interleave boosts the Disk Index 50 percent.

```
C:\>si c:
SI-System Information, Version 4.00, (C) Copr 1984-87, Peter Norton

        Computer Name:  IBM/PC-AT
     Operating System:  DOS 3.20
   Built-in BIOS dated:  Friday, June 20, 1986
       Main Processor:  Intel 80286              Serial Ports:  2
         Co-Processor:  None                   Parallel Ports:  0
Video Display Adapter:  Monochrome
   Current Video Mode:  Text, 80 x 25 Monochrome
Available Disk Drives:  6, A: - F:

DOS reports 640 K-bytes of memory:
   288 K-bytes used by DOS and resident programs
   352 K-bytes available for application programs
A search for active memory finds:
   640 K-bytes main memory     (at hex 0000-A000)
    47 K-bytes display memory  (at hex B000-BBC0)

  Computing Index (CI), relative to IBM/XT: 11.2
       Disk Index (DI), relative to IBM/XT: 3.2

Performance Index (PI), relative to IBM/XT: 8.5
```

Index and Disk Index equate to in the real world. Golden Bow Systems provides a better performance picture with their *VBENCH* program, part of the *Vopt* optimizing package. Figure 6-3 shows the various performance aspects of an 8MHz 80286-based machine running with one memory wait state. The standard of measurement is a plain 6MHz IBM PC AT.

By increasing the speed of the central processing unit (CPU) chip, you can radically alter your computer's performance. You cannot run your processor faster than your memory chips will allow, however. Figure 6-4 shows the same system as in Figure 6-3, but after installing zero-wait state RAM and increasing the CPU speed to 10MHz.

Not all performance tweaks are without trade-offs. Figures 6-3 and 6-4 show the video performance of a SunTek monochrome graphics card. As shown in Figure 6-5, installing the ATI EGA-Wonder card drastically cut the video speed. This video board emulates both EGA and Hercules graphics, so the increased flexibility

Figure 6-3. Golden Bow System's *Vopt* package includes the VBENCH (VisualBenchmark) performance measurement program.

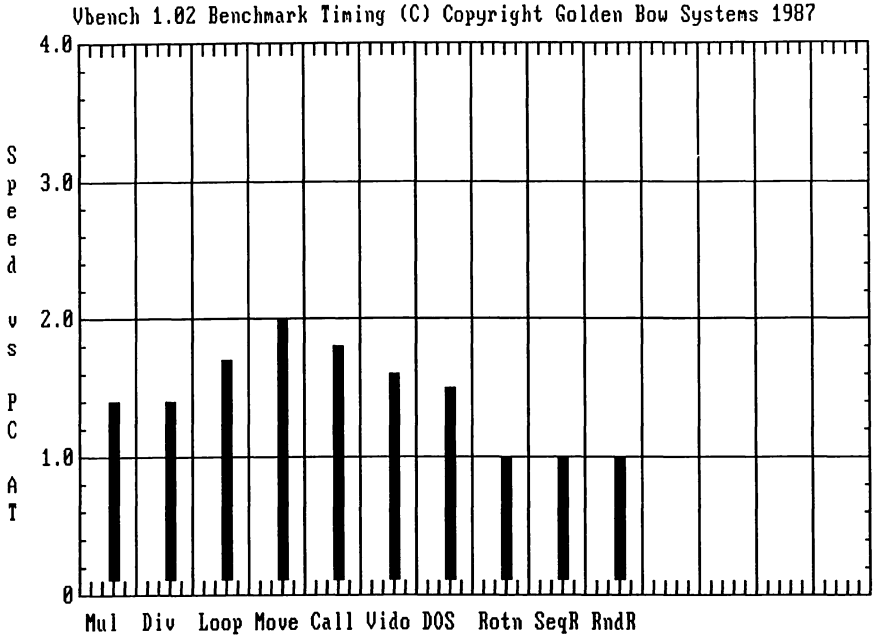


Figure 6-4. VBENCH shows the impact of increasing clock speed to 10MHz and using no-wait memory.

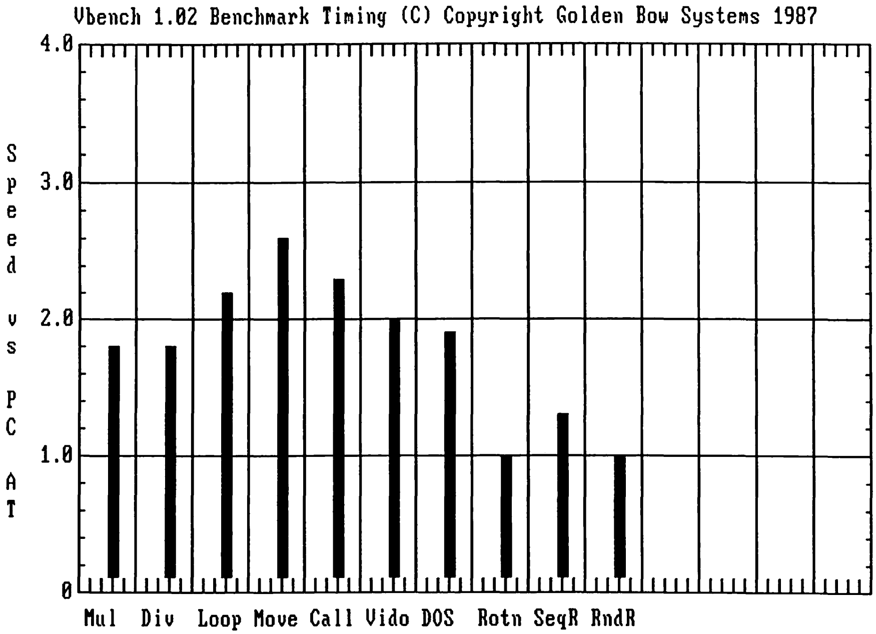
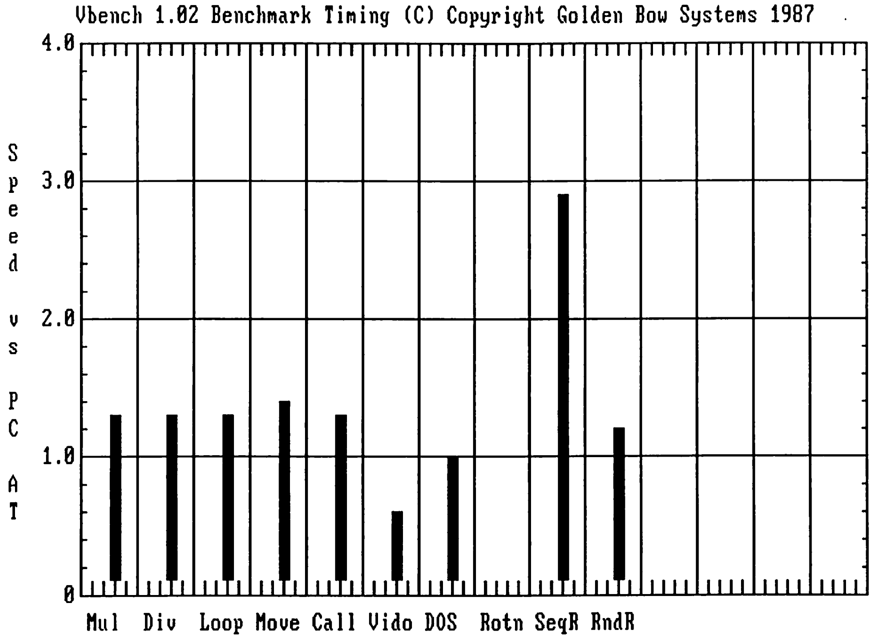


Figure 6-5. As indicated by VBENCH, a disk cache dramatically increases sequential disk reading.



costs video speed. Installing a disk cache using an AST Rampage board increased sequential file access (SeqR), but the slow RAM chips forced dropping the processor speed to 8MHz. The *VBENCH* screen indicates the impact of performance trade-offs.

Core International, manufacturers of high-performance disk drives and the *COREfast* backup software, have developed a set of benchmark programs. *PC Magazine* developed another test suite. All these products are available free on CompuServe, as well as benchmark tests from various user groups and computer bulletin boards.

Faster Disks with RLL

Run length limited (RLL) disk controllers store disk data in a format that is more condensed than that presently used in PCs. The RLL format squeezes 50 percent more data onto a disk. As a by-product of higher data density, RLL units transfer an equivalently

greater amount of data in the same time frame. Therefore, switching to an RLL controller increases disk data transfer speed by 50 percent.

But there are great dangers in this process. If you're considering such a move, carefully explore three factors:

- Is your hard disk adequate for RLL? Most low-cost drives don't have the accuracy needed to drop their error tolerances by 33 percent. Check with the controller manufacturer or vendor to find out if your drive is capable of working within these tight tolerances.
- Does your drive work with this controller? Even if your drive theoretically can handle it, not all drives work with all controllers.
- Can you afford disk problems? Realistically, if tolerances go down radically, the chances of alignment problems or misreading data increase. You will have to back up your files even more faithfully. In the event of a failure, can you afford to be without your hard disk for days at a stretch? If you run completely parallel systems (generic software sites) as discussed near the end of Chapter 5, the answer may be yes. For most of us, the answer is no.

Advertisements give the impression that merely inserting an RLL controller into your system will boost disk capacity, turning your nearly-full 20Mb drive into a 30Mb unit. It isn't so. Even if you are lucky enough to have a drive that works with RLL, your existing data is in the standard format. You must copy your entire disk (preferably two sets of file-by-file backup), install the new controller, perform a low-level hardware format of the drive, partition for a larger MS-DOS area, reformat for MS-DOS, and then restore your files. Chapters 3 and 4 explain these tasks; they are not difficult, but they are more involved than merely inserting a board into a slot.

Although RLL technology is well-established on minicomputers, it is new technology for micros, and new technology is risky. But technology marches on, and the bugs are disappearing from RLL systems. Don't assume anything. Ask questions and make sure you believe the answers.

Faster Processing

The heart of every personal computer is a single computer chip: the microprocessor. Its job is to read and execute the instructions that make up the programs you run. Each member of the IBM PC family is associated with a specific chip. The 80286 chip, found in the IBM PC AT, is faster than the 8088 chip that drives IBM PCs. The 80836 chip in PS/2 Model 80 systems is faster still.

Coprocessors are supplemental chips which take over specific tasks from the main processor. Most common in the PC family are math coprocessors. These chips, typically designated 8087 or some similar number such as 80287, specialize in fast arithmetic computations.

The system clock, based on a crystal oscillator, controls processor timing. The system clock is the computer equivalent of a metronome. Metronomes don't do anything but tick. Each time the metronome ticks, a musician plays one beat. Likewise, each tick of the crystal signals the processor to advance one step in reading and executing instructions. Like a metronome, system clocks can have varying speeds. As a metronome ticks faster, musicians play correspondingly faster, if possible. As clock speed increases, processors speed up.

With this information, you can understand the three possible ways of upgrading processor speed:

- Speed up the system clock
- Upgrade to a faster processor chip
- Add supplementary processors

Does it make sense, on your particular machine, to try any of these? Maybe. Hardware boards and add-ons are available for each. Prices range from under $100 to over $1,000. Many accelerator boards (as they are usually called) include other features such as additional memory, graphics, and software.

It pays to upgrade if you want the speed and your system can take advantage of the improvement. Be aware, though, that even the hottest processor is useless if it doesn't work on your machine.

- Speed advantages are relative. A board manufacturer may claim 100 percent improvement in processor speed. If your memory, busses, and disk are limiting you, there may be only a fifteen or twenty percent improvement.

- Speed can be dangerous. Pushing a processor to its limits can cause intermittent errors as the system gets ahead of itself. Just think of a piano player trying to keep up with a wildly ticking metronome.
- Changing hardware can change the way your software works. If a program's pace is tied to the system clock, speeding up the clock can send the program racing across the screen as a useless blur. Changing or adding processors can cause weird side effects ranging from jumpy screens to actual processing errors.

IBM PC AT systems are either 6MHz or 8MHz 80286-based computers. They insert one wait state (tick) during each memory access to allow the use of slow, inexpensive versions of dynamic RAM (DRAM) chips.

In the competetive marketplace, speed has become a selling point. Many clone systems offer no-wait memory access. If your system allows disabling the memory wait, do it. Table 6-1, below, lists the recommended DRAM speeds for various CPU speeds with and without a wait state. You need to run a faster processor (measured in megahertz, MHz) with faster memory, measured in nanoseconds (ns).

Table 6-1. Typical Dynamic RAM Speed Requirements

CPU Clock Speed	RAM Wait States	Access Speed
6MHz	1	200ns
6MHz	0	120ns
8MHz	1	150ns
8MHz	0	100ns
10MHz	1	120ns
10MHz	0	80ns
12MHz	1	100ns
12MHz	0	80ns

Increasing Memory

The most cost-effective performance booster on most personal computer systems is a hard disk. Increased Random Access Memory (RAM) is a strong second. The reason is partly the same: Prices have dropped so dramatically that both are now great bargains. One megabyte of RAM, which ten years ago cost roughly $10,000,

costs about $100 today (of course this price doesn't include the circuitry necessary to use the RAM).

Adding RAM boosts performance by decreasing disk accesses. Since RAM access is typically ten times faster than disk access, reading or writing anything from RAM saves time. Increasing your RAM can let you:

- Fit large programs and data files into memory without frequent disk swapping
- Provide more buffers, increasing disk read/write efficiency
- Create ramdisks or disk caches, both discussed in the next section
- Create bigger data files
- Allow more RAM-resident tools to smooth and speed system use.

There are three main types of RAM:

- System memory
- Expanded memory
- Extended memory

You can have as much as 640K of system memory (also called *conventional memory*) in your PC. This is the plain vanilla workhorse memory where you normally run your programs. System memory is cheap. If you have less than 640K, or 512K at the very minimum, go out today and buy more. Memory comes on inexpensive add-on boards, often bundled with other features such as battery-operated clocks, extra I/O ports, and utilities. It's easy to install: Just open your system cover, plug the board into an empty slot, and (if necessary) set a switch or two.

Extended memory is more complicated. It makes tricky use of a MS-DOS anomaly. Both 8088 and 8086 processors, the heart of the PC and XT computer families, can address up to one megabyte (1Mb) of memory—just over one million bytes. This is divided among Read Only Memory (ROM), the 640K of system memory, and RAM accessible only to MS-DOS.

The top 64K of MS-DOS's memory is unused unless you have an Enhanced Graphics Adapter. Hence, you can add this 64K to your system memory, making up to 704K of system memory. Public domain machine language programs are available that make use of this extra memory.

Even if you have an EGA, a PC XT has a 64K window from E0000-EFFFF. Both the PC XT and a PC AT have a 64K window

from D0000-DFFFF. Any of these windows can be treated as a bank-switching window:

- Lotus, Intel, and Microsoft developed the standard for using this memory, called the *LIM standard,* after their corporate initials. It's also called the *EMS,* or Expanded Memory Specification. The original LIM standard breaks expanded memory into 16K pages, which it moves into and out of the 64K window as needed. It's possible to address and use up to eight megabytes of memory through this 64K gap. The LIM standard limited the memory to data buffering.
- AST Research created a superset of LIM memory, called Enhanced EMS (EEMS). This scheme switches 64K blocks anywhere within the base 1Mb of memory. It also allows noncontiguous blocks to switch. Quarterdeck's *DESQview,* discussed in Chapter 5, makes effective use of EEMS memory.
- In August 1987, EEMS became adopted as the hardware configuration for the version 4.0 EMS standard. Adding a device driver into CONFIG.SYS further extends EEMS to allow running programs from anywhere within a 32Mb memory space.

Unlike the 8086 and 8088 chips, the 80286 and related processors in the AT family computers can directly address up to 16 megabytes of memory. The 80386 can directly address two billion bytes (two gigabytes, or 2G) of memory. Both processor families use the first megabyte of memory just as an 8088 or 8086 does. At such times, they are in *real mode,* emulating an 8088/8086 in every way. The remaining memory, called *expanded memory,* is only available in *protected mode,* a special processing mode unavailable on 8088/8086 machines. Therefore, expanded memory is sometimes also called *protected memory.*

Ramdisks

A ramdisk is exactly what it says: a pseudo-disk drive created in RAM. Ramdisks are also called *virtual disks* because they exist only in memory, lacking true disk hardware. However, they appear just like a real drive to MS-DOS. Ramdisks can speed your system tremendously.

What belongs on a ramdisk? Ideal candidates are programs or data files that are:

- Small (unless you have expanded or extended memory)
- Read only
- Frequently used

Examples include:

- Program overlay files
- Tutorial files when you're in a learning mode
- Program support files, such as help files or spelling lists
- Batch files
- Small, frequently used program files

Ramdisks can go in expanded or extended memory if available. File size is not an issue if extensive RAM is available. Program files loaded only once don't belong in ramdisk. You gain nothing on a single load.

Frequently used batch files are particularly appropriate for ramdisk. They're small and very slow on hard disk. That's because MS-DOS must reread the disk separately for each and every line in the batch file. You may not think they're slow, but you'll be amazed at the difference when you move them to ramdisk. Likewise, if there are a few MS-DOS external commands you use often, try putting them in ramdisk. You can move both the batch files and MS-DOS command files into ramdisk either of two ways:

- Copy them one at a time.
- Put all your ramdisk batch files in a separate directory, then copy the entire directory contents. For example, you might create a \BATCH subdirectory, \RAMDISK, containing the desired files. After creating the ramdisk drive D, fill it with the single command:

  COPY C:\BATCH\RAMDISK*.* D:

  Put this command into AUTOEXEC.BAT if appropriate.

Temporary work files are another good candidate for ramdisk. Although they're not read only, it doesn't matter if you lose them. Read-only files are important because ramdisk storage is only temporary. If your system crashes, locks up, or dies for any reason, everything on your virtual drive will be lost. Read-only program files

are no problem: Just reload them. But if you've spent the last three hours building the ultimate spreadsheet in RAM, you will probably feel some anguish as the system goes down.

Once you experience ramdisk speed, it's tempting to put data files there. The best advice is:

- Don't take the chance unless you're willing and able to reconstruct the file, if necessary.
- Copy the file to disk periodically in any case.

Some programs, such as *WordPerfect*, make periodic backups automatically. The user can adjust the frequency of automatic backups. If you use *WordPerfect* with a ramdisk, make sure the backups go to hard disk, not ramdisk. Other programs, such as *WordStar*, let you write files to disk from within the program. In the worst case, if you can't save to disk within the program, have your batch file save the data when you exit the program.

Below is a sample batch file that copies *WordStar* overlay files and a user-specified data file into the ramdisk, then copies the data file back to its original hard disk directory when you're done working on it. Note one side benefit of such a batch file: It lets you edit *WordStar* files anywhere on your disk, thus bypassing the single directory limit of *WordStar* version 3.3.

```
ECHO OFF
COPY C:\WS\*.OVR D:
CD C:\%1
IF EXIST %2 COPY %2 D:
D:
C:\WS\WS %2 COPY %2 C:\%1
C:
ECHO ON
```

This batch file expects two parameters: the data file directory and data filename. It expects to find the *WordStar* program (WS.COM) and overlay files on directory C:\WS and assumes the ramdisk is drive D. Upon exiting *WordStar*, it copies the data file back to its original location.

A typical call might be

```
WS WS\DATA MYFILE.DOC
```

Nonvolatile Ramdisks

Two commercial products greatly improve the value of ramdisks:

- JRAM Combo Pack software from Tall Tree Systems
- nonvolatile expanded memory boards

The JRAM Combo Pack is a $60 software kit. It includes a ramdisk driver, print spooler, and the *Jet* fast copy program. It is specifically mentioned here because it lets you create a ramdisk that isn't destroyed by pressing Ctrl-Alt-Del or a reset button (a *warm boot*). (It is not immune to a *cold boot*, accomplished by turning the power off and on.) Those who use ramdisks regularly will understand how much frustration this simple feature can avoid. If it lets you safely move just one data file to disk without loss, it's worth the price.

Nonvolatile RAM means it doesn't lose data when your system loses power. A nonvolatile RAM board uses a separate power cable to keep the board active. It includes a battery backup to protect against power failures. One example is the X2C Expanded Memory Board from ABM/Franklin Computer Systems. It costs around $600, but may be worth it if you're serious about using expanded memory.

Creating a Ramdisk

You must use special software to create a ramdisk. This software can be:

- A device driver loaded by your CONFIG.SYS file
- A separate software program you run from the MS-DOS command line

You can buy stand-alone ramdisk software. Also, most expanded or extended memory boards include ramdisk software designed to work with the board. MS-DOS 3.0 and 3.1 include a ramdisk driver called VDISK.SYS. For MS-DOS 3.2 or later, it is called RAMDRIVE.SYS.

Ramdisk software varies in flexibility. The best let you choose:

- Ramdisk size
- Sector size
- Cluster size

- Number of directory entries
- Where to put the ramdisk: in system, expanded, or extended memory

Disk size should be as large as you can make it without cramping other applications. If you are using extended or expanded memory, your ramdisk can be very large. On system memory, size depends on your applications. If you have large program files with little disk access, you will have little or no need to use a ramdisk. If you are using small program files with extensive disk access, you will have many uses for as large a ramdisk as you can manage.

Reflex is a perfect example of a program which should not use a ramdisk. First, the entire program loads into memory without overlays or support files. Second, whenever it works on a data file, *Reflex* loads the entire file into memory. A ramdisk would yield nothing, but would reduce the available system memory, hence reducing the maximum data file size.

WordStar 3.3, by contrast, is an ideal ramdisk application. It uses multiple overlays and extensive data swapping. It may read each overlay several times in the course of a single word processing session and move data sections on and off disk repeatedly. The more RAM space disk you can make available, the better.

The ideal sector size and cluster size are hard to determine and don't matter much in any case. In general, larger sectors are faster. If you have many small files, then small clusters—say, 1K—waste less space. Sectors that are 512 bytes are probably just fine. Directory size is important to avoid wasting space. One 512-byte sector holds 16 directory entries. Don't ask for more than 16 entries unless you really need them.

Typically, the ramdisk receives the next available drive identifier. If you have two floppy disks and a hard disk, the ramdisk will be drive D. You can make as many ramdisks as you want, by repeated driver loads or program calls.

Once you create your new drive, copy onto it the files you want to access from ramdisk. Then direct your programs to look there, using PATH and APPEND or by logging the ramdisk as the current drive.

Using VDISK

The MS-DOS RAM drivers are all many of us will ever need for ramdisk operation. Most commercial programs produce significantly faster ramdisks. However, the MS-DOS drivers are inexpensive; they're easy to use; and they work. You simply load a driver in CONFIG.SYS. The format is:

DEVICE=[drive][path]VDISK.SYScapacitysectorsize dirsize /where
DEVICE=[drive][path]RAMDRIVE.SYScapacitysectorsize /where

You can put comments between each value. Omitted values adopt defaults. Table 6-2 shows acceptable and default values. A sample VDISK call, using VDISK off of the UTIL subdirectory and creating a 100K ramdisk in system memory, might be:

DEVICE=C: \UTIL \VDISK.SYSsize=100sectorsize=512directory size=16

Table 6-2. MS-DOS RAMDisk Driver Specifications

Specification	Minimum	Maximum	Default	Acceptable Values
Disk size	16K bytes	available memory - 64K	64K	Any, but memory is assigned in 16K chunks.
Sector size	128 bytes	512 bytes	128	128, 256, 512 (1024 on some systems)
Directory size	2 entries	512 entries	64	
/where				/E = extended memory /A = LIM/EMS memory

Disk Caching

Another excellent use of extra RAM is through disk caching software. Caches are essentially buffers. They minimize disk accesses by trapping disk I/O requests and copying each data sector as it moves to or from disk.

When a disk read is requested, the caching software checks whether the requested data is already in the cache. If so, it supplies the data without the necessity of reading from disk. MS-DOS often requests that recently read disk sectors be reread. VCACHE, supplied with the *Mace+Utilites*, shows this in its status display in Figure 6-6.

Figure 6-6. Even a small disk cache will save many disk accesses to speed system performance.

```
C:\>cache-em

Vcache 1.16 version for the Mace+Utilities
Copyright (C) Golden Bow Systems 1985-1987
Licensed for use on a single system only.

Already loaded - Cache size  512 kb

Options: P T

Statistics:  1018 read requests -  57% satisfied from cache
              261 write requests -  4% duplicates (ignored)
                0 memory errors     0 disk errors

             1.2% of computer time in read-write requests as of 12:43:17

C:\>
```

Ordinarily, you would write an entire file to disk even though only a few sectors have changed. Caching software checks each sector being written to disk to see whether it has changed since it was last read or written. Only changed sectors are rewritten. Since disk writes take even longer than disk reads, reducing disk writes can dramatically improve performance.

Golden Bow System's disk cache program, *VKETTE*, boosts disk access remarkably. However, it seems to pause before updating directories so the user can sometimes corrupt floppy disks by removing them too soon after being written to. Avoid caches for removable media unless you can force an immediate directory update.

Sooner or later a cache fills up. One important function of cache software is deciding what to keep and what to remove from the cache. MS-DOS buffers use a *Least Recently Used (LRU)* strategy. Whenever the MS-DOS buffers are full, the buffer which has been unused for the longest time is overwritten. LRU is an acceptable strategy. However, sometimes a *Least Frequently Used (LFU)* strategy is better. Most caching software uses a combination approach. For this reason and others, even though caches are theoretically very similar to MS-DOS buffers, a cache system is nearly always more efficient than MS-DOS buffers.

All cache programs require a certain amount of overhead, both

for the software itself and for recording usage statistics on each sector stored in the cache. However, if you use the cache constantly you can eliminate your extra MS-DOS buffers and thus free nearly all the space required for program overhead.

Commercial Cache Software

Features worth looking for in a caching program include:

- Small program overhead: The smaller the program, the more room you have for the actual cache.
- Small housekeeping overhead: Some programs use much more RAM than others to record usage.
- Easy installation, without the need to understand all features immediately.
- User-specified cache size.
- Statistical reporting on the number of disk accesses avoided.
- Choice of system, expanded, or extended memory.
- User-controlled locking: The ability to lock files or disk sections into the cache.
- Ability to save and restore cache status: Even if you turn off your machine overnight, you can start with the same usage configuration tomorrow.
- Ability to cache both reading and writing of floppy disks.
- Full-track reads and writes.

Probably the most important feature to verify is that the program writes data to disk without a long delay. A few caching programs, including some in the public domain, hold data an excessively long time before writing.

Caching software is sometimes part of larger packages. For example, the *Mace Utilities* includes the VCACHE caching program in Figure 6-9. However, stand-alone packages are the rule. Two of the most popular at present are *Lightning*, from the Personal Computer Support Group, and *Flash*, from Software Masters. Table 6-3 compares features in the two programs (as of this writing—features change frequently). In general, *Lightning* is smaller and therefore better for use on systems with less RAM; *Flash* has more advanced features. Also, for very large caches *Flash* is actually smaller than *Lightning* because its housekeeping is more efficient.

Other commercial products include:

- *QUICKcache* from MSD
- *SuperPC-Kwik* from Multisoft (at about $80, a fast favorite of many)
- *Fast Forward* from the Mark Williams Company

Table 6-3. A Comparison of *Flash* and *Lightning*.

Feature	***Flash***	***Lightning***
Program size	6K	16K
Overhead required for a 500K cache	20K	50K
Maximum cache size	1.5Mb	1.7Mb
Can use expanded memory	Yes	Yes
Can use extended memory	Yes	No
User-controlled data locking	Yes	No
Ability to save and restore cache status	Yes	No
Buffered reading of floppy disks	Yes	Yes
Buffered writing of floppy disks	Yes	No
Success reports available	Yes	Yes
Full-track reading and writing	Yes	No

Ramdisk Versus Disk Caching

Should you use a ramdisk or a disk cache? Disk caches are usually the better choice:

- Disk caches speed all processing, not just specific applications. Often the File Allocation Table and major directories are among the data in the cache. This reduces access time for every file. Also, some caching programs read or write entire tracks at once. Since most disk access time is spent locating data and positioning the heads, full-track I/O speeds processing for all files larger than a single cluster.
- Disk caches work on all files, not just files small enough for the ramdisk. Even if an entire file cannot fit in the cache, its most frequently accessed sectors can fit. Small pieces from a number of different files can co-reside in the cache.
- Unlike ramdisks, disk caches are safe for all data. There is no danger in reading data from disk, since it can always be reread if the system crashes. Writing data is nearly as safe. The software holds the data in RAM just long enough to make sure it really needs to

be written—seconds, at most. The disk write occurs as soon as checking is complete.

- Once you install the cache, results are automatic. You don't need to worry about loading, saving, and reloading as you do with a ramdisk.
- Even a small 30K or 50K disk cache should bring noticeable improvements, while a ramdisk this small might not even be useful. Remember also that if you always load the disk cache, you can reduce the number of MS-DOS buffers, thus freeing more memory.

Optimizing Your Interleave

The interleave factor was discussed in Chapter 3. Interleave refers to the layout of sectors on each track. Rather than putting sector two right next to sector one, most drives interleave at least one additional sector between them. This gives the system time to process sector one before it's time to deal with sector two.

A higher interleave is more conservative than a lower one. It allows more time to make sure the system is ready. If the interleave is higher than needed, you are wasting time, forcing extra revolutions every time you read or write a track. If the interleave is too low, the situation is even worse. You will need an extra complete revolution for each and every sector.

Since the ideal interleave depends on your system's overall processing speed, it makes sense that any changes to your system could change the optimal interleave. You should check the interleave and consider correcting it if you've changed the clock speed or if you've made other speed-related changes.

Several programs, including the *HOPTIMUM* program in *Mace*'s HTEST/HFORMAT package, can test your disk and tell you the ideal interleave. They temporarily format a portion of the disk with various interleaves, as indicated in Figure 6-7. Using read-only testing still reformats the affected region, so make sure you have two complete, trustworthy backups of the entire disk.

Since the ideal interleave often proves to be much better than what you're using, you'll probably want to reformat the disk. If you've been using the disk for a while, this can be a scary proposition. You'll be wiping your disk completely clean and starting over from scratch, just as in Chapter 3. Is it worth it? It depends on how inefficient your present disk interleave is. Disk read/write speed is

Figure 6-7. *HOPTIMUM* tries various hard disk interleave factors to determine the best for your processor speed and drive type.

```
C:\DIAGNOSE>hoptimum

HOPTIMUM - Hard disk interleave OPTIMIZER Program  -  Version 1.0.4
Copyright (C) [1986,1987] Kolod Research, Inc.  All rights reserved

Drive - C - will be formatted and tested as:

            read/write heads:   4
                   cylinders: 614
                     sectors:  17

Formatting parameters will be:

           beginning surface:   0 (surfaces start internally at 0)
              ending surface:   3
          beginning cylinder: 603 (cylinders start internally at 0)
             ending cylinder: 613
           sector interleaves   1 through    7
                machine type: PC/AT (or compatible)
             READ/WRITE test: -READ-
           TRACK/SECTOR test: -TRACK-
```

so basic to your system's performance that the chance to make a significant improvement is probably worth the effort.

If you do decide to try reformatting, the steps are:

- Verify that you have two complete, readable file-by-file backups. The key to successful reformatting is to have good, reliable backups. Don't even consider reformatting unless you have two complete backups which you *know* to be reliable. File-by-file backup will ensure contiguous files when you reload. The next section explains why this is useful.
- Run *HTEST* or another thorough disk testing program. Keep a list of all bad sectors it reports, as shown in Figure 6-8. You'll need this later to make sure all bad sectors get remarked after reformatting.
- Run *HOPTIMUM* or another interleave tester if you haven't already.
- Run a controller-level formatting program such as *HFORMAT* to establish the new interleave (as discussed in Chapter 3), test the disk, and mark all bad sectors. Thorough sector testing and marking is extremely important. After all, why bother going through

this process if you're going to end up with an unreliable medium? Keep a list of all bad sectors marked.

- If there are bad sectors identified by *HTEST* which are not recognized by *HFORMAT*, mark them. These are marginal sectors that might fail at any time. Be safe rather than sorry. If you're using *HFORMAT*, you can manually enter the sector numbers. If not, you'll need to use further sector testing/marking programs like *Norton*'s DT (DiskTest) before loading all your software.
- Run the MS-DOS FDISK program to set up the boot table and establish your MS-DOS partition. Chapter 3 gives details.
- Run MS-DOS FORMAT to create the file allocation table and root directory and prepare the disk for logical MS-DOS operation. Again, Chapter 3 gives details.
- Run an MS-DOS disk testing program, such as *Norton*'s DT, as a final check for bad sectors. Then restore your files from the backup.

Figure 6-8. HTEST reports bad sectors so you can map them out if you reformat the drive.

```
       Kolod Research, Inc.      - HTEST -       IBM PC Hard disk TEST

  Current Cylinder:  613 Current Head:    3  Error page #:    1  of    1

 Cylinder   Head   R/C      Cylinder   Head   R/C      Cylinder   Head   R/C
 --------   ----   ---      --------   ----   ---      --------   ----   ---
    336        0   0Ah         379        0   0Ah         419        0   0Ah
    443        3   0Ah         612        0   0Ah

ENTER . . . (A)-Exit HTEST
            (N)-Next error page (P)-Previous error page  (H)-Error HELP text
```

Some experts feel that reinterleaving puts your disk at risk by introducing bad sectors or other problems. Others disagree, arguing that if you remap the sectors that were excluded during the manufacturing process, the potential gains make it imperative to consider reinterleaving. Currently available commercial software is adequate to the task. Just make sure you use competent software, thorough testing, and safe, reliable backups.

Reorganizing for Increased Efficiency

Reorganizing actually breaks down into two different activities:

- Defragmenting files
- Putting files in their most advantageous positions on the disk

Defragmenting files. When MS-DOS writes a cluster to disk, it first checks the file allocation table (FAT) to find the next available cluster to use. If one file is several clusters long, MS-DOS repeats the process for each cluster. There is no guarantee that the various clusters making up a file will be contiguous. This doesn't matter to MS-DOS, but it matters to you: Repositioning the disk's read/write heads for each noncontiguous cluster slows disk access appreciably.

With a new, empty disk, there is no problem. Because all clusters are available, the next available cluster is always adjacent to the last cluster. If MS-DOS used cluster 101 for the last write, it will use cluster 102 for this write and cluster 103 for the next. However, the longer you use your disk, the less this is true. As you erase files you create holes in the FAT—available clusters situated at random positions around the disk.

Large files created gradually also result in noncontiguous clusters. This would happen if you built up a large mailing list file over a period of many months. Each time you add enough new names to require an additional cluster, MS-DOS stores the addition in the next available cluster from the FAT. However, the new cluster is probably nowhere near the rest of the file. You end up with one file spread all over your disk. When MS-DOS reads the file, it must reposition the heads on different cylinders and tracks for each separated cluster.

A badly fragmented disk is not only slow, it's dangerous. For example, some backup programs won't run if data is too spread

out. The disk drive simply can't keep up with the backup program's demands if the read/write heads must reposition repeatedly.

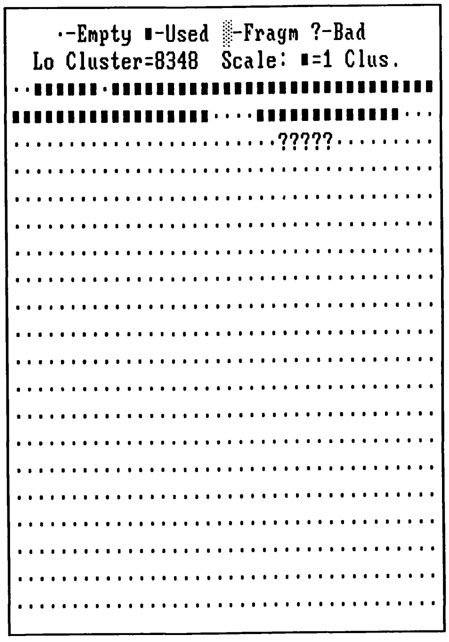
This is where *defraggers* come in. These are programs that reunite files. They may also eliminate FAT holes in the process. The end result is a compact sequence of files, written to contiguous clusters.

A number of defraggers are available. *Disk Organizer* from Soft GAMs Software is probably the most widely used shareware reorganizer. Some of the stand-alone commercial products are *Vopt* from Golden Bow Systems, *Disk Optimizer* from SoftLogic Solutions, and *DS Optimize* from Design Software. Some products, such as *Mace Utilities* and *Power Tools* from MLI Microsystems, include reorganizational programs as part of larger packages.

Sometimes the most effective strategy includes using multiple tools for the same task. A nice feature of the *Vopt* package, shown in Figure 6-9, is that it only moves files that are quickly reorganized. This contrasts sharply with the more complete, slow processing of the *Mace Utilities.* Consider including *Vopt* in your system startup batch file. Run *Mace* during periodic house-cleaning.

Figure 6-9. Golden Bow System's *Vopt* only defragments files that can be moved quickly, so it works well within the system startup batch file.

```
Vopt Version 2.01  serial no. 702025
Copyright (C) Golden Bow Systems 1987

39600 sectors total disk space
  136 sectors in 30 directories
33372 sectors in 814 user files
  160 sectors marked bad
 5932 sectors available on disk

    1 empty files
    5 clusters in 1 fragmented files
 8420 is the highest used cluster
 3954 is the lowest empty cluster
    7 empty clusters in the used space

4 files (15 clusters) were moved
No directory sectors were updated
7 seconds to compact disk
```



One other defragmenting option is open to you: You can reorganize with a brute-force file-by-file backup-and-restore. This may be a reasonable option if you have a tape backup system or other dedicated backup. The backup-and-restore process creates the desired contiguous files. However, for best results you should reformat the disk after copying and before restoring, which is a scary proposition. And, more important, why take the risks inherent in backup-and-restore when safe defragging programs cost less than $50?

> Warning: Unless you specifically know your reorganizer can handle them, you should uninstall copy-protected programs before reorganizing. Some protection schemes are tied to specific disk locations or organizations; moving files and reorganizing data will make the program unidentifiable to the copy-protection scheme. Programs which require a key disk should be safe; others probably aren't safe.

Here's what to look for in a defragger:

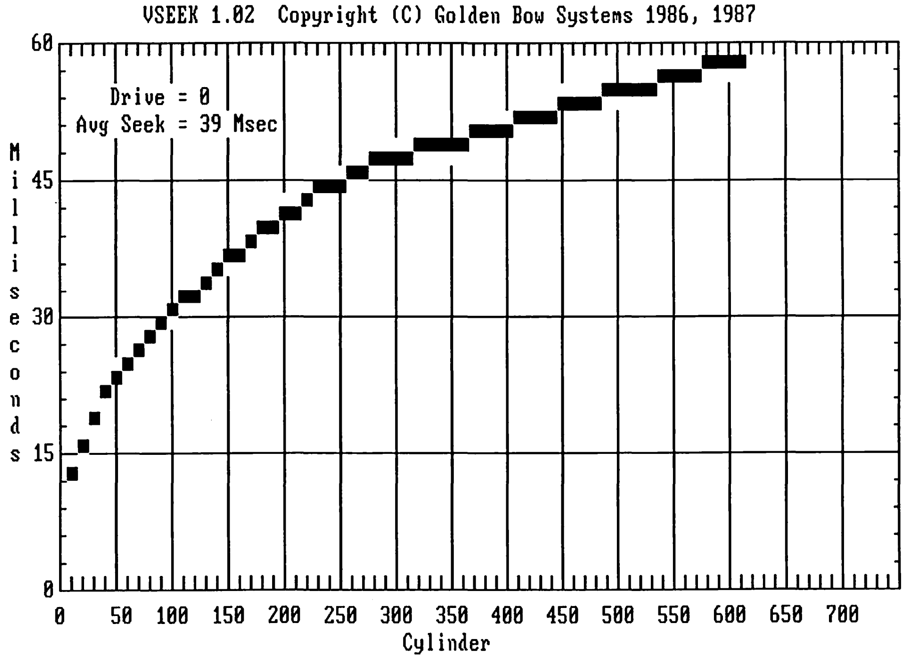
- Speed: Some programs take hours to restore a badly fragmented disk, others take minutes.
- Safety: Most programs copy files in a safety-first mode—the old copy isn't erased until the new copy is safely in place. There is always a slight danger of the system crashing between writing the FAT and writing the directory entry, but you can use CHKDSK or *Norton Utilities* to recover lost clusters if need be. Of course, it's always wise to have a complete, reliable backup before starting an extensive reorganization.
- Flexibility and user control: Can you reorganize individual files or directories? Many programs can only process the entire disk. Of course, if the program is fast enough, this is no problem. Can you specify how thorough a job to do? *Disk Organizer* lets you decide whether to defragment completely or just to combine chunks that don't require massive data movements.
- Checking: Is there a checking mode which tells you the degree of fragmentation without actually moving data?
- Exclusion: Can you exclude specific programs from the reorganization process? If so, this is a solution to the copy-protection problem mentioned in the box above.

Rearranging your disk. Even if all your files are contiguous, there's another reorganizational step which will enhance your hard disk performance. It's simple in principle, but often hard to execute: Put your most frequently read files near the front of the disk (low-numbered cylinders near the FAT and root directory) and put the most frequently written files on the higher-numbered cylinders near the end.

As you can see from Figure 6-10, disk reads are faster close to cylinder zero. The root directory, from which all searches begin, is located in the first cluster of the MS-DOS partition because it is the first file written during the initial formatting. It makes sense to put directories, batch files, overlays, programs, and other frequently-read files as close to it as possible.

In Chapter 4, you read how to set up your disk this way initially. However, over time things change, and it makes sense to go back for periodic cleanup. Also, after you've used your disk awhile, you'll probably add a number of data files which you'll update regularly. Putting these after your other files will reduce future disk fragmentation.

Figure 6-10. Files near the front of the disk are found more quickly than those toward the end.



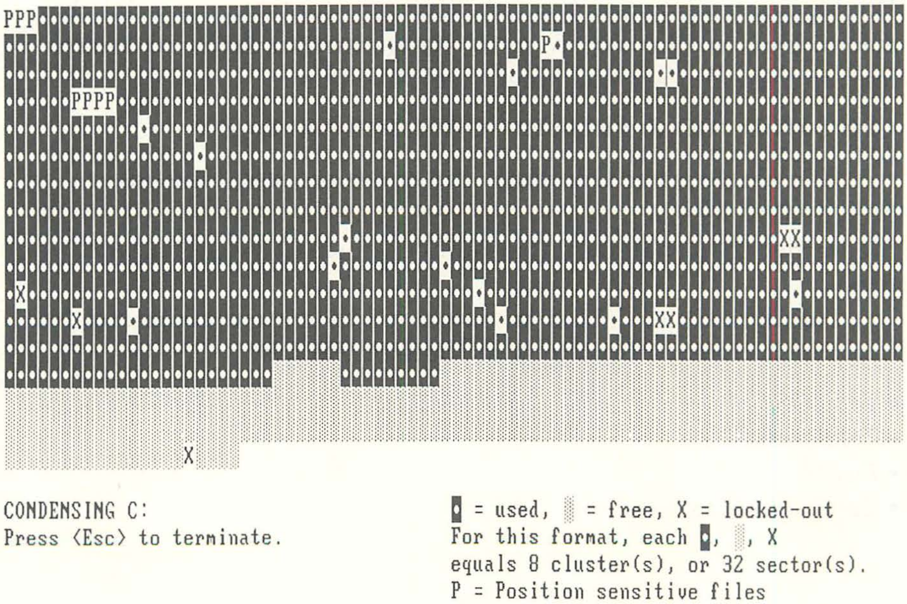
Many defragmenters will also help you rearrange your disk, if they allow you to specify the most-read and most-written programs or directories. *Disk Organizer* has an option to write all fragmented programs at the end of the disk. A fragmented file may well be an updated file, so using this strategy continually will tend to move frequently-written files to the end of your disk.

The CONDENSE program in *Mace Utilities* has a clever design feature which is useful here. It rewrites disk files in the following sequence:

- Root directory files
- Subdirectories
- Hidden, read-only, and system files
- All other files, in their original sequence

By marking all read-only files, such as program files, with the read-only attribute, you can insure that these files move to the front of the disk, which is where you will want them, because read-only files are typically frequently read files. Also, since they rarely change, keeping them at the front minimizes reorganization time. As you can see in Figure 6-11, a comprehensive reorganization can take quite a while.

Figure 6-11. The comprehensive reorganization done by *Mace Utilities* can take over an hour per disk drive.



Sleek System Software

The programs discussed in this section have one common purpose: enhancing your use of MS-DOS. These programs make MS-DOS faster, sleeker, easier to use, more complete. Although categories frequently overlap, they fall loosely into the following groups:

- Command editors
- Path extenders
- Fast file managers and MS-DOS shells

Command editors. Command editors let you edit the MS-DOS command line just as you would edit any other file. You are not confined to the current command line: You can scroll through previous lines, just as you would through a text file. When you reach the desired command line, you insert, delete, and replace text until you get just what you want. Then you execute the line by pressing Enter.

CED (Command EDitor), a public domain program written by Chris Dunford, is one of the original MS-DOS command editors. Synonym lists are a favorite feature of *CED* and other command editors. Synonyms are essentially macros, created online or kept in ASCII text files. You can create a long single- or multiline command, then identify it with a shorter synonym. Within *CED*, for example, a caret (^) separates command lines within a synonym. Double quotes (") surround any synonym containing special characters. Thus, a *CED* synonym to change to your *Reflex* data directory and list all data files might look like this:

CED SYN D "CD C:\REFLEX\DATA^DIR *.RXD /P"

The single letter *D* initiates the entire process. One advantage of synonym lists is that you can keep many synonyms in a single file, rather than maintaining a number of small, wasteful batch files.

Handy as it is, *CED* is far surpassed by current commercial programs. Here's a brief look at some of the best:

- *TopDOS*
- *TallScreen*
- *PCED*
- *Command Plus*

TopDOS, from Frontrunner Development, is particularly easy to use. Unique among editors discussed here, it cannot handle multiline synonyms. However, it includes a macro processor and the program offers many handy features. For example, the cursor changes appearance to signal that you're in *typeover* rather than *insert* mode. One command transposes letters (changing *ODS* to *DOS,* for instance). You can *autocomplete* an MS-DOS command by pressing a single key, once you've typed enough to uniquely identify the command. You can sort directories by file size, extension, or creation date. The program lists for $69.95 and uses 128K RAM.

TallScreen, from Qualitas, Inc., requires about 64K of RAM and costs $49.95. In addition to the usual features, it offers exceptional editing capabilities. You can edit anything onscreen or in a buffer. You can mark a block of text, and then copy or print it. You can set default operating preferences. The program pops up anywhere, and also offers pop-up help for both itself and MS-DOS.

PCED is an upgraded commercial version of *CED*. It comes from the Cove Software Group, lists for $38, and uses as little as 16K RAM (*CED* uses 25K). *PCED* synonyms can use the current date, time, drive, path, and more. This makes it easy, for example, to restore the current directory after completing some side task. You can even save this information for next time. As a result, you can, for example, create a synonym to backup everything since the last backup date. You can create synonyms for each user, then log their computer activity automatically. Using predefined commands, you can customize your directory, change file attributes, or pass keystrokes into your programs as a keyboard macro program would. You can call up *SideKick* from within a *PCED* synonym. With a few useful exceptions, synonyms work only at the MS-DOS command line so you don't have to worry about accidentally executing them within a program.

Command Plus from ESP Software Systems is different from the other programs described here. It replaces COMMAND.COM. It starts by including all standard MS-DOS commands plus the command editing and scrolling features discussed above. Then it goes further: It adds syntax checking, online help, and additional switches and options for many commands. For example, a single delete command can delete not only a directory, but all its subdirectories. Instead of the frustrating and limited TYPE command,

the BROWSE command displays files in useful chunks, one screenful at a time, and includes a string search option. *Command Plus* also includes SCRIPT, a powerful block-structured command processing language. You can use integer and string variables, For-While loops, GoTo's, subroutines, case statements, and assignment, Boolean, conditional, and mathematical operators. You can also access the system environment, perform string operations, and position characters on the screen. This is a quantum leap beyond most command editors. *Command Plus* lists for about $80.

Even more commercial command editors are available. In addition, a growing number of MS-DOS shells, including *Hot!* and *Norton Commander,* include features described here.

File finders/path extenders. *File finders,* also described more accurately as *path extenders,* make it easy to find any kind of files anywhere on disk. The PATH command provides a list of directories to search, but only for .BAT, .EXE and .COM files. Path extenders use the same path instructions to search for all other kinds of files. The most common is *File Facility* from IBM's Personally Developed Software series. Also called *FileFac,* this is a bargain at about $20. It uses 2K of resident memory. You can change paths at any time with the PATH statement and can use environment variables to keep two paths active at once. You can define one set of read-only paths and another set for reading and writing.

A predecessor of *File Facility, SEARCH* from ArborSoft, a shareware program, also combines PATH and APPEND. It works with MS-DOS version 2.0 or later. As you can see from Figure 6-12, SEARCH offers a single path for all files. It can be deactivated and subsequently reactivated without forgetting its defined path.

Other path extenders include:

- *DPath* (or *DPath20*), a simple shareware program
- *SmartPath* from Software Research Technologies
- *FilePath* from SDA Associates
- *DPATH Plus* from Personal Business Solutions

Among the useful features found in some programs are

- An option to override MS-DOS paths with a different set
- Options defining exactly what kinds of files to search for
- Manual override by including an exact path as part of a filename

Figure 6-12. ArborSoft's *SEARCH* combines PATH and APPEND into a single shareware program.

```
C:\UTIL>search /h

                         SEARCH 1.0

SEARCH is a general purpose replacement for the DOS PATH command.
SEARCH works with arbitrary programs on arbitrary files. SEARCH
is distributed on a user-supported basis. Type SEARCH /L for
licensing information.

Available functions are:

    Set...................... SEARCH [d:]path[[;[d:]path]...]
    Clear.................... SEARCH;
    Display.................. SEARCH
    Activate................. SEARCH /A
    Deactivate............... SEARCH /D
    Edit (with function keys). SEARCH /E
    Help..................... SEARCH /H
    License Information...... SEARCH /L

Copyright (C) 1984  Arborsoft Systems, Inc.
                    P.O. Box 4599
                    Ann Arbor, Michigan 48106
```

DPATH Plus may be the most complex path extender available as of this writing. It is also the most difficult to use effectively. However, its power is fascinating. It allows you to

- Change directories while a program is running
- Specify that all files matching a given wildcard go in a certain directory
- Emulate the JOIN command, so files in all directories on all paths show up when you do a directory listing of the current directory

DPATH Plus's list of features points up the potential dangers of path extenders. First, you may have multiple files with the same name scattered throughout your disk. You must be sure you know what you're working with when you start widening your path from a single track to a boulevard. Second, you may find yourself writing files to a different directory than you intended. Path extenders help you find files but aren't very good about knowing where to write them. You must make sure each file ends up where it belongs by methodically checking everything the path extender does.

Fast file managers and DOS shells. *Fast file managers* are great favorites among hard disk users. They solve the contradiction between the inherent power of a hard disk system and the inherent frustration of the limited, clumsy operations of MS-DOS.

Fast file managers are frequently called *shells* because they surround MS-DOS, offering menus or prompts to simplify the most frequent MS-DOS commands. However, good file managers go far beyond MS-DOS, offering features unavailable in MS-DOS. People sometimes think shells are for new users, to protect them from the difficult syntax of MS-DOS. But more advanced users appreciate them even more, because they understand the need so fully. That's why we've included them in this chapter on speed, rather than in the previous chapter on ease of use.

A good file manager should have all or most of the following features:

- Point-and-shoot file and directory selection
- File tagging by individual selection, wildcard, date, or file attribute
- Mass file operations on tagged files: move, copy, delete, rename, or change attributes
- Directory operations: create, delete, rename, sort files by name, date, extension, or attributes
- File viewing, editing, and printing, including scanning, scrolling, and DEBUG-type display options for non-ASCII files
- File find or wildcard matching anywhere on disk
- Overall statistics on number of files, directories, and free space
- Technical information about your drive, such as the number of tracks and cylinders
- System status: available memory and CPU, port and drive information

Notice that almost none of these functions is available directly through MS-DOS. Many file managers include additional MS-DOS functions such as FORMAT. Others give you help in building MS-DOS commands, either prompting you for correct syntax or letting you pull filenames from a display window into the command. At the very least, nearly all let you execute MS-DOS commands directly from within the program. So, you have nothing to lose and everything to gain by using a good DOS shell.

Some file managers incorporate other features such as menu builders, macro processors, desktop accessories, backup programs,

file encryption/decryption, and more. It pays to look at all these features when selecting the best program for you. Also, and equally important, make sure the program interface comfortably matches your workstyle. Most interfaces fall loosely into the following groups:

- Collections of separate stand-alone utilities
- Menu-driven programs, either Lotus-style, pull-down, or traditional
- Command-driven programs starting from a main directory tree

Some file managers are RAM-resident pop-up programs; most are not. Although you can usually execute programs, MS-DOS commands, batch files, or BASIC programs from within them, they may be too large to keep resident when running large programs. Memory requirements range from 50K to 200K and prices go from $30 to $100. Execution speed is another major variable, although most are very fast.

Figure 6-13. The *X-Tree* Startup Display.

```
Path: \
┌────────────────────────────────────────────────┬──────────────────┐
│ \                                              │FILE: *.*         │
│ ├─BATCH                                        ├──────────────────┤
│ ├─CIE                                          │DISK: C:          │
│ │  ├─3-D                                       │ Available        │
│ │  ├─CCPM                                      │   Bytes: 1,671,168│
│ │  ├─DEMOWARE                                  ├──────────────────┤
│ │  ├─FONTS                                     │DISK Statistics   │
│ │  ├─GRAPHICS                                  │ Total            │
│ │  ├─GSX                                       │   Files:    1,469│
│ │  ├─SEER                                      │   Bytes:16,676,811│
│ │  └─TOOLS                                     │ Matching         │
│ ├─DOS                                          │   Files:    1,469│
│ ├─DV                                           │   Bytes:16,676,811│
│ ├─HDWR                                         │ Tagged           │
├────────────────────────────────────────────────┤   Files:        0│
│ AUTOEXEC.BAT      IBMBIO   .COM                │   Bytes:        0│
│ BACKUP   .M_U     IBMDOS   .COM                │ Current Directory│
│ COMMAND  .COM                                  │  \               │
│ CONFIG   .SYS                                  │   Bytes:  118,619│
└────────────────────────────────────────────────┴──────────────────┘
DIR       Available  Delete  Filespec  Log disk  Makedir  Print  Rename
COMMANDS  ^Showall  ^Tag  ^Untag  Volume  eXecute
 ↑↓ scroll  RETURN file commands   ALT menu        F1 quit F2 help
```

As an example, here is a brief look at *X-Tree*, by Executive Systems. *X-Tree* covers nearly all the above listed file manager features and more. When you first bring up the program, *X-Tree* displays five windows, as seen in Figure 6-13:

- System statistics, including available space, and the number of directories and files
- Directory window containing a directory map for the logged disk
- File window listing files on the current directory
- *Filespec window* showing the initial file selection criterion, *.*
- Command window listing available commands

You have a great deal of control over *X-Tree*'s display. Within the directory window, highlighting a directory displays its files in the file window. You have three available file display formats, from condensed to complete. You can:

- Expand the file window to fill the whole length of the screen
- Change the file specification at any time, causing only matching files to display
- Highlight and tag individual files
- Tag all files matching the current *filespec*
- Select files based on attributes: hidden, system, archive and read-only
- Remove or display the command window
- Show and work with all files on the disk matching the current filespec
- Choose among five different directory sort criteria

Figure 6-14 shows a full-screen *X-Tree* file window with the most complete display format, including file attributes. Online help is available, although it is not context sensitive. However, the program is remarkably easy to learn and use. Once you've used it awhile, it's hard to remember how you lived without it.

There are many fast file managers currently available. Each has different strengths, so it's possible you'll want more than one. Most cost around $50. Table 6-4 briefly lists some of the well-known file managers and their outstanding features.

Figure 6-14. *X-Tree* expands its directory window to fill the screen and can display the file attributes.

```
Path: \BATCH

1-2-3    .BAT      162 .a..  ERADIR  .BAT      204 .a..  FILE: *.*
ALLDIRS  .BAT       93 .a..  FKEY10  .BAT    1,083 .a..
ALLTSR   .BAT       39 .a..  FKEY10  .MNU    1,863 .a..  DISK: C:
ALPHA    .BAT      625 .a..  GEM     .BAT      108 .a..   Available
APPTSR   .BAT       78 .a..  HGCSNAP .BAT       97 .a..     Bytes: 1,671,168
BACKUP   .BAT      538 .a..  KERMIT  .BAT       71 .a..
BATCH    .DSN    4,402 .a..  LIST    .BAT       24 .a..  DIRECTORY Stats
CLR      .BAT       55 .a..  MACE    .BAT       43 .a..   Total
CONFIG   .SYS      207 .a..  NEWCHILD.BAT      533 .a..     Files:        56
COPYDISK.BAT       173 .a..  NUMKEY  .BAT    1,018 .a..     Bytes:    22,999
CPM80    .BAT      219 .a..  NUMKEY  .MNU    1,863 .a..   Matching
CWD      .BAT       21 .a..  OVER    .BAT      188 .a..     Files:        56
DIRS     .BAT       78 .a..  PAINT   .BAT       68 .a..     Bytes:    22,999
DIRS     .SN     3,827 .a..  PRNSETUP.BAT    2,048 .a..   Tagged
DOSTSR   .BAT       29 .a..  PROTECT .BAT        9 .a..     Files:         0
DOWN     .BAT      186 .a..  RESTORE .BAT      547 .a..     Bytes:         0
DTG      .BAT       96 .a..  RETURN  .BAT      121 .a..   Current File
DV       .BAT       87 .a..  ROOT    .BAT      106 .a..     1-2-3    BAT
ERA      .BAT        8 .a..  SAYWHERE.BAT       43 .a..     Bytes:       162

FILE      ^Attributes ^Copy ^Delete Filespec Log disk ^Move ^Print
COMMANDS  ^Rename ^Tag ^Untag View eXecute
←↑↓→ scroll  RETURN dir commands   ALT menu        F1 quit F2 help F3 cancel
```

Table 6-4. Some Well-Known File Managers

Product and Vendor	List Price	Outstanding Features
1 DIR Plus, Bourbaki, Inc.	95.00	Exceptional customization features with multilevel menus, system information, good design for new and experienced users.
Direc-Tree Plus, Micro-Z Co.	49.50	Easy to learn and use, menu and macro generators, can attach notes to filenames, *WordStar* subdirectory support.
dirWORKS, Keep It Simple Software	25.00	Simple pop-up menus, extremely friendly can display system information; excellent disk formatting option.
DOS2ools, E-X-E Software Systems	99.00	Outstanding collection of 43 well-designed stand-alone utilities. Optional menus provide a smooth learning curve.
KeepTrack Plus, Finot Group	79.00	Tree-oriented program. Includes thorough backup and restore.
Norton Commander, Peter Norton Computing	75.00	Completely or partially memory-resident, good customization and menu-building, thorough manual.

Product and Vendor	List Price	Outstanding Features
PathMinder, Westlake Data Corp.	49.95	Lotus-style menus, 5K virtual mode, COMPOSE command to pull highlighted files into MS-DOS commands, encryption, expert mode, good universal file display.
QDOS II, Gazelle Systems	44.95	Fast processing, good built-in editor, Lotus-like menus.
SNAP, The Mt. Whitney Group	99.95	Menu-driven or point-and-shoot, encryption and decryption, *WordStar*-like editor, good global delete, online MS-DOS tutorial.
X-Tree, Executive Systems Inc.	49.95	Simple design, good user control, visual tree.

Keyboard/Screen Controllers

One disadvantage of getting faster is that suddenly little slowdowns you never noticed before become irksome. For example:

- Slow cursor movements
- The beep MS-DOS gives out when you overflow the 16-keystroke type-ahead buffer
- The need to turn down your screen's brightness when you go to lunch (or risk burning next year's financial projections into your screen)
- Flickering screens and blinking cursors

One small but welcome group of utilities addresses these problems, plus others that arise with your keyboard and console.

KeyBuff is a simple 1K RAM-resident public domain program whose sole function is to expand the MS-DOS type-ahead buffer. That's enough. If none of your other software provides this function, get your hands on *KeyBuff*.

KBFIX2, a public domain program from Skip Gilbrech, includes a type-ahead buffer. It also displays toggle key status, makes Scroll Lock and CapsLock work as you'd like, and makes cursor movements faster and better.

NoBlink/Accelerator, from Nostradamus, lets you turn the blinking cursor into a solid block. It also gives you three cursor speeds to choose from, and you can easily reset the speed in every application.

Flicker Free, from Gibson Research, works with any display adapter that uses the 6845 CRT controller chip. Hercules, CGA, EGA cards all work from three to 15 times faster with this software. It only costs about $40.

Fansi-Console, from Hersey Micro Consulting, integrates a number of keyboard and screen features, including a much improved ANSI.SYS device driver. Screen scrolling is six times normal speed. It eliminates snow and flicker on a CGA. You can retrieve material that's already off the screen from the screen buffer, and you can use EMS. *Fansi-Console* also offers an expanded type-ahead buffer, fast cursor movements and stops, and even a simple macro generator. Unfortunately, it uses nearly 40K of RAM, has a hard-to-read manual, and costs $75. Still, many people love it.

Cruise Control, from Revolution Software, also radically speeds the cursor and offers *antiskid braking*: As soon as you lift your finger, the cursor stops. It offers three control strategies—different operational modes to work with various software. The default strategy covers most cases. It can properly perform screen-dimming for both text and graphics modes, restoring the display when you press any key. Figure 6-15 displays its help screen. *Cruise Control* costs $29.95.

Figure 6-15. Cruise Control provides date and time data entry, stops cursor overrun, and prevents screen phosphor burn.

On-Line Help Panel			
CRUISE	Cruise Control Panel	[5][+]	Faster Cursor
CRUISE/{A-C}	Control Strategy	[5][-]	Slower Cursor
CRUISE/{1-60}	Auto-Dimmer Delay	[5][Delete]	Dim Display Screen
CRUISE/0	Disable Auto-Dimmer	[5][Key]	Cruise With [Key]
CRUISE/X	Remove From Memory	[+]/[-]	Alter Cruise Speed
CRUISE/R	Redefine [5] Key	[5][Tab]	Control Strategy
CRUISE/W{1-8}	Select Repeat Delay	[5][*PrtSc]	Auto-Dimmer On/Off
CRUISE/#{1-8}	Select Date Format	[5][D]	Insert Date
CRUISE/@{1-6}	Select Time Format	[5][T]	Insert Time
CRUISE/S{1-8}	Select Cursor Speed	[5][Insert]	Program On/Off

Cruise Control Version 3.02	
Auto-Dimmer delay is 5 minutes.	Control Strategy is A.
The [REV] key definition is [5] on the numeric keypad.	
For help type: CRUISE/H [Enter]	August 16, 1987 10:29 PM

Both *Fansi-Console* and *Cruise Control* dim your screen after a user-specified time elapses with no keyboard input. If you don't use *Cruise Control* or another program with this feature, you'll want *ScrnSave,* a 1K public domain program that dims your screen after two minutes without keyboard input, then restores it instantly after any keypress. Its only drawback is that it's incompatible with graphics-mode programs on Hercules graphics cards.

RAM Managers

There are several ways to remove a RAM-resident program without having to reboot your computer. First, though, you must understand how MS-DOS uses RAM.

Programs load into RAM in first-come, first-served sequence: Earlier programs have lower memory addresses. Ordinary programs release their RAM automatically upon program completion. *Terminate-and-stay-resident* (*TSR*) programs stay loaded in RAM. These are so-called RAM-resident utilities such as *SideKick* and so many of the programs discussed in the last few chapters. All programs must be removed in *last in, first out* (*LIFO*) sequence. Removing an earlier program simply leaves a hole in RAM which is useless at best and possibly dangerous.

A growing number of TSR programs include built-in removal options. For example, the /X option in *Cruise Control* removes it from memory (assuming you've loaded it last, of course).

TurboPower Software's *TSR.ARC* offers another solution. This is a collection of TSR-management programs they developed and release to the public domain. It is available on the Borland SIG of CompuServe. MARK and RELEASE, are a matched pair of programs. MARK marks a memory location and saves all the interrupt vectors; RELEASE releases everything above the last MARKed location. If you know you'll want to remove a TSR program you

- Mark memory
- Load the TSR program
- Release when you want to remove the program

Figure 6-16 shows the TSR list in memory. This display is generated by MAPMEM, part of the *TSR.ARC* package.

Figure 6-16. MARK/RELEASE/MAPMEM are public domain programs to manager TSR applications.

```
C:\>mapmem

Allocated Memory Map - by TurboPower Software - Version 2.1

 PSP  blks bytes owner    command line       hooked vectors
----- ---- ----- -------- ------------------ ------------------------------
0008   1   83280 config
1C93   2    3232 command                     22 24 2E
1D6B   2    1552 MARK     alltsr
1DCE   2     464 MODE     N/A                14 17
1DEE   2   13360 CACHE-EM 512 /p/t           13 19
2181   2    1584 MARK     dostsr
21DD   2    2224 HBIOS    0
2260   2   83216 FRIEZE   PSCRIPT S1 0 1HE... 10 21 46 ED F0 F4 F6 FC
36A9   2    8928 HGC2PCX                     05
38CF   2  422672 free

block   bytes   (Expanded Memory)
-----   ------
    0   524288
 free  1245184
total  1769472
```

MARK and RELEASE have one disadvantage. If you use them in a batch file, as is typical, RELEASE removes the program but doesn't actually free the memory until you leave the batch file. These programs are part of TSR.ARC, an archive file available on the Borland SIG of CompuServe.

PopDrop from InfoStructures is a $19.95 commercial product which provides MARK/RELEASE service. Typically, you load *PopDrop,* which is itself RAM-resident, before loading any other TSR programs. Before each new TSR layer you include a **POPDROP UP** statement. To remove a layer, you **POPDROP DOWN. POPDROP C** removes all TSRs except PopDrop itself, and **POPDROP S** removes even that.

Referee, from Persoft, goes further in theory. You can specify a different list of RAM-resident utilities to load with each application. *Referee* takes care of the rest. However, some users report problems running program combinations under *Referee* which operate properly under *PopDrop*. The program retails for $69.95.

This brings up another danger with RAM-resident programs: Not all programs work well together. As your RAM gets more and

more crowded, you'll need to pay more attention to manufacturers' instructions. Some programs must load first, some last, and some are simply incompatible with each other. Sometimes it's handy just to see what's where. *VOPT* from Golden Bow Systems includes a simple program, *VTSR*. As seen in Figure 6-17, VTSR displays what's where in RAM, how much RAM each program uses, and what memory markers are in place.

Figure 6-17. *VTSR* displays TSR memory usage and memory markers.

Vtsr Version 1.01 TSR Info Copyright (C) Golden Bow Systems 1987

SizeKb	Name	Uses/Comments
1.4	MARK	MARK PARAMETER BLOCK FOLLOWS
4.9	PRINT	Ints(15,28,2F)
.3	MODE	Ints(14,17)
21.9	CACHE-EM	Ints(19) (C) Golden Bow Systems 1985-1987
1.4	MARK	MARK PARAMETER BLOCK FOLLOWS
16.0	CED	
63.2	SK	Ints(08,09,13,16,1C,20,25,26,27)
2.0	HBIOS	
81.1	FRIEZE	Ints(10,21)
8.5	HGC2PCX	Ints(05) (C) 1987 Controlled Information Environm

```
<Sun 6-14-1987 23:43:09.42>
C:\DIAGNOSE>
```

Many Small Solutions Equal One Big Solution

The last major topic for this chapter is actually a number of small topics. The overall advice is: watch for shortcuts. Three or four small timesavers can yield as much as one big one. The biggest single change in moving to a hard disk is the ability to work easily with a number of powerful programs, either together in an integrated job stream or separately, quickly moving from one unrelated program to another. Therefore, this chapter ends with a number of small shortcuts related to this multiprogram environment:

- Translators
- Print spoolers
- Task-switching

Translators. If you keep all your data on your hard disk, accessing a single data file from two programs is a snap. But can both programs actually read and use the data? Often the answer is no. That's where translators come in. Translators change formats but not file contents, so different programs can use your data.

Two major types exist. One type works primarily on numeric or database data. Typically, the program translates the data into one of several standard formats such as Lotus format, IBM's DCA format, or Data Interchange Format (DIF).

The other main type translates text. The major word processors each use different control codes to indicate underlining, boldface, and so on. *WordStar* adds high-order bits which cause havoc in some other programs. Straight ASCII files must have both high-order bits and control codes removed.

Many high-level programs now include built-in translators, at least to bring data in if not to ship it out.

For word processing files, try *Software Bridge* from Systems Compatibility. It converts among nine word processors, preserving margins, tabs, headers, footers, footnotes, italics, soft hyphens, and nearly everything else. At $149, it's less than the cost of many translators servicing only two programs, and it does a better job than most.

Print spoolers. Say that you've just completed that 90-page report, and as it prints you'd like to run your billing system to prepare an invoice. But your system is completely involved with waiting for the printer, page by agonizing page. You're ready for a print spooler.

In essence, a print spooler is simply a printer buffer plus its controlling software. Even the fastest printer is far slower than your computer. Rather than wait for the printer to work its way through an entire document, a print spooler saves the material in RAM or on disk, then feeds it to your printer whenever your system is idle. Many printers include built-in RAM buffers, but often they are only two to four kilobytes. A print spooler will save files hundreds of kilobytes long.

Many unrelated utilities throw in a print spooler as an extra, and there are also a number of public domain spoolers. If you are in the market for a print spooler, you might consider looking at *PrintQ* from Software Directions. It costs about $90. It allows you

to start printing from any page, print up to 255 copies automatically, and view documents without printing. You can also check or change printing status at any time using a pop-up window. Since *PrintQ* always saves files on disk, not in RAM, you can even print them another day.

Task-switching. The ability to switch programs in midstream—multitasking—is a major feature of environment managers, which were examined in Chapter 5. If you want multitasking but don't want to go to the extent of buying an environment manager, consider a task-switcher. They don't offer windowing or the special features of the environment managers, but they do let you switch quickly from one program to another.

Software Carousel, from SoftLogic Solutions, lets you load as many as ten *program contexts*. A program context is usually a single program, but it could include affiliated pop-up programs. For instance, your word processing context might contain a spelling-checker or thesaurus. Pressing a single key moves you from context to context. The swapped-out context goes into a suspended state. The effect is similar to the pause button on a tape recorder: The program stops, but its status is preserved exactly, ready to restart when you tell it to. Saving and restoring data files is done for you. *Software Carousel* swaps to expanded or extended RAM if available, or to your hard disk, if necessary.

DoubleDOS, from the same manufacturer, solves a slightly different problem. You run only two programs, but you run them both at once. This is called *concurrency*. The background task doesn't stop processing, but simply runs more slowly, using time when your main program is idle. You can make one program invisible or, if you have both color and monochrome displays, display one program on each console. The usefulness of such a program depends heavily on your environment: Do you have the kind of background tasks to use such a system, and is your system fast enough to run two programs without noticeable slowing?

Conclusion

This chapter points out some of the ways you can make your hard disk system faster and more powerful. From hardware to software to little extras, there is much you can do to boost performance.

Furthermore, *better performance* doesn't have to mean *much more expensive*. You can buy 640K of RAM, plus a high quality file

manager, MS-DOS command editor, ramdisk software, and disk cacher for under $300. Throw in a few public domain programs to handle print spooling, TSR management, screen dimming, and cursor control. If you do, you'll see a noticeable improvement in the ease, speed, and pleasure of using your system.

To help you sort out the mass of possible improvements, use this Performance Booster Checklist.

System ______________________________

Checklist completed by (name) ______________________________

Date ______________________________

	Program we currently use to perform this function:	*Is it important for us to upgrade our software for this function?* No	*Yes*	*Very*
MS-DOS Command Editing				
Command processor language				
Command buffer with scrolling				
Command editing				
Synonyms				
Command syntax checking				
Online help for MS-DOS commands				
File Management				
File tagging by				
Wildcard				
Date				
Attribute				
Single file				
Mass operations on tagged files:				
Copy				
Move on the same drive				
Move to another drive				
Rename				
Delete				
Individual file operations:				
Find				
View				
Edit				
Non-ASCII edit				
Print				
View/change attributes				
Directory Management				
Visual directory tree				
Directory listing by				
Date and time				
Name				
File size				
Extension				
Multidirectory search/display				

Desktop Processing

Calendar				
Clock				
Calculator				
Appointment scheduler				
Notepad				
Database manager				
Phone lists				
Phone dialer				

Miscellaneous

Ramdisk software				
disk cache				
Multilevel user menus				
Macro generator				
Path extender				
Interleave test/recommend				
Low-level disk formatter				
File defragmenter				
Checking mode				
Systematic file sequencing				
RAM (TSR) manager				
System information display				
Drive information display				

Keyboard, Screen, and Printer Control

Enlarged type-ahead buffer				
Cursor speed control				
Screen dimmer				
Cursor blink control				
Print spooler				
Flicker/snow protection				

Multitasking

Task-switching				
Concurrency				

Other valuable features in products we are considering: (backup software, encryption, protection features, logging, other)

Chapter 7

Security

Security

In the good old days, sensitive data was kept in locked rooms staffed by computer personnel 24 hours a day. Now, your sensitive files may sit unattended on top of your desk. When you worked on floppy disks, you could at least lock up your files at night. But what do you do with your hard disk?

This chapter examines valid reasons for protecting data, and the kinds of dangers to protect against. You'll see software methods of access restriction, from batch files and menu shells to file encryption, passwords, and full-scale software security systems. You'll also have a look at hardware security via keys or removable media.

> Guard now against loss, damage, or theft later:
>
> - Protect people from themselves.
> - Give people access only to what they need.
> - Match protection to the level of sensitivity.
> - Consider both hardware and software protection.
> - Set up a system.

You should be concerned with data security if you:

- Share a system
- Have data someone else might want to see
- Are using programs someone else might want to copy

What to Protect and Why

To many of us, *security* brings up images of foreign spies stealing national secrets. In fact, computer security means protecting your system from:

- Accidental damage
- Unauthorized viewing or copying of restricted data
- Unauthorized copying of programs
- Malicious damage

If you don't lock your front door, anyone can walk in. Locking the door makes it harder. The world's best security system can keep almost everyone out, but still can't insure complete safety. Likewise, you should never relax and think your computer system is secure. However, some reasonable precautions make sense.

What You Can't Touch, You Can't Hurt

A young child can find countless ways to get into trouble. No one can possibly guard against all of them. However, you can take obvious precautions by putting matches, guns, and caustic cleansers out of reach.

Likewise, there are many ways to harm your files. The only possible defense is complete, timely file backup. Because this is so important, an entire chapter is devoted to it. Meanwhile, you can guard against the most obvious dangers. This discussion won't cover deliberate misuse of your system or data. Rather, you will see how to protect against avoidable, accidental damage.

> Keep potential danger safely out of the way.
>
> - Remove or rename potentially hazardous programs.
> - Control inexperienced users with menus.

The first danger is commands or programs which move, copy, or change entire files without adequate verification. MS-DOS FORMAT and COPY commands are the worst.

Suppose you want to format a new floppy. You're operating off your hard disk, drive C, as usual. You've done this many times before, so you're not paying much attention. In fact, you're talking on the phone as you take out the new floppy and put it in drive A. You type the MS-DOS command **FORMAT**. You see the MS-DOS prompt *Press any key to begin formatting drive C*. You press the space bar. By the time your phone call is over, your entire hard disk, drive C, is reformatted. Everything is erased.

Protecting Against the FORMAT Command

The safest way to handle the FORMAT command is to rename it and use it only through a batch file. It's named FORMAT.COM and should be in your MS-DOS directory. Use the MS-DOS RENAME command to change it to, say, XFORMAT:

CD C: \DOS
REN FORMAT.COM XFORMAT.COM

Make a batch file containing the command to format floppy disks in drive A.

C: \DOS \XFORMAT A:

Save it in your UTIL directory under the name FORMAT.BAT.

If you sometimes format in drive B, make another batch file in the UTIL directory, called FORMATB.BAT.

C: \DOS \XFORMAT B:

Protecting Yourself from the COPY Command

If you tell MS-DOS to COPY file A to directory B, and directory B already contains a file named A, COPY overwrites it. Nothing will remain of the original file A in directory B. The computer will never stop to ask if that's really what you want to do. It doesn't care if it's replacing a later file with an earlier one, or even if the two files are completely different.

> Make it easy to avoid copying over the wrong file.
>
> - Name files carefully and consistently, as discussed in Chapter 5. There's no excuse for two different kinds of data having the same name.
> - Use COPY only as part of batch files. You can enter the filenames correctly once and not have to worry about future careless errors.
> - Don't use COPY at all. Use XCOPY (from MS-DOS 3.2) or use the copying features from one of the many commercial hard disk utilities.

Other Dangerous Commands

SYS, FDISK, and RECOVER are three other examples of potentially dangerous commands. Since they're rarely used, you may want to remove them entirely from your hard disk. Keep copies of these files on floppy disks, of course.

Another option is to move them to a less accessible directory. Instead of keeping them in the DOS directory, put them in a special subdirectory that's not part of your normal PATH.

Hide or Change Your File with DOS File Attributes

The *hidden attribute* keeps inexperienced users from even knowing a file exists. You can use disk utility programs such as *Norton Utilities, PCTools,* and *Xtree* to turn on hidden status. However, hidden files can still be overwritten. Furthermore, a knowledgeable user can simply turn off the hidden attribute and view the file using the TYPE command.

More useful is the *Read-only attribute.* Recent versions of MS-DOS can turn on this bit, as can the disk utility programs listed above. People can view your data, but they can't damage it. You can create a simple batch file to turn off Read-only status immediately before using the file, then reset it afterward:

```
ATTRIB -R myfile
myprogram
ATTRIB +R myfile
```

> Use hard disk subdirectories for security.
>
> - Make it easy for people to go where they belong.
> - Make it hard for them to go anywhere else.

Once again, check that your disk organization matches your needs. Each application area should have its own directory. Each user's files should be separate. If two people use a single application, each should have his/her own data directory (if the program allows it) so they won't share the data.

Use batch files, menu programs or MS-DOS shells to provide smooth access to applications. Even without passwords, a simple menu program will keep new users on track (unless they want to get into trouble). Don't put MS-DOS on the menu unless users need it and can handle it. Consider giving them batch files for specific functions, if needed, rather than letting them run amok with the power and dangers of MS-DOS.

Keep People Honest

Until now, the discussion has been aimed at minimizing the chances for accidental damage. The next topic of discussion is keeping people away from data they shouldn't be seeing. You may have

specific concerns about specific people. More likely, you're simply observing good preventive business practices. Don't make it easy for one angry employee to destroy months of work.

> Be different. Be unique.

Think of ways to slow down a snoop. If you use a keyboard control program like *SmartKey* or *Superkey,* consider changing some of the keys. For example, substituting | for the Enter key will frustrate anyone unaware of your little trick.

Some experts recommend using bizarre, meaningless filenames. This trick would be more likely to hurt you more than it hurts potential criminals, but it might start you thinking.

> Move sensitive files onto floppy disks.
>
> - Keep sensitive data on floppy disks permanently.
> - Copy sensitive files to the hard disk only while you are working with them.
> - Remember that erased files can also be unerased. Obliterate sensitive files with Central Point Systems' *WIPEFILE* program.
> - Control all media containing sensitive data. File or shred old reports and ribbons.

If each set of related files fits on a single floppy, and if it's not too slow working with them on floppy disks, just keep them there. This does not entail any special hardware or software for security (except the cost of a secure place to store the floppy disk).

If you want the speed advantages of hard disk data, or if the files are too cumbersome for floppy disks, move them onto the hard disk for the actual time you're working with them. A single batch file can

- Copy the data from floppy to hard disk.
- Run the program(s) using it.
- Copy the updated data back onto a different floppy (Chapter 8 explains why you always copy onto another floppy).
- Remove all traces of the files from the hard disk.

When MS-DOS deletes a file, it does not physically remove it. Instead, it changes its directory pointer. Physically, the file remains

right where it was. If you're protecting against deliberate wrongdoing, MS-DOS DEL or ERASE is not enough. You must use a disk utility program to physically erase the file from the disk. These programs either *scrub* the marked-as-deleted directory entries, or they write blank information (as FORMAT does) to the disk areas occupied by the file.

Even if you're keeping your data on floppy disks, you will want to physically erase any working files or backups created on the hard disk. Also remember to clean out your print buffer, if you're using one.

Don't leave sensitive printouts lying around your desk or trash. Even the ribbon that printed the report is dangerous. Anyone who can read backwards can reproduce whatever you printed.

File Encryption

Moving data back and forth is cumbersome. If you elect to keep data files on your hard disk, consider *encrypting* them.

Encryption means changing a file by using a coding system. The file is unreadable until it is decrypted or decoded using the same system. Many utility programs, including *Superkey*, *SmartKey*, *PathMinder*, and *DiskOptimizer* offer encrypt/decrypt commands.

Encryption uses a password you provide. You can't decrypt the file properly without the password. If you forget the password, you're stuck. It would be a good idea to keep a decrypted backup copy of the file under lock and key somewhere in case you forget your password.

Encryption will stop casual viewing but may simply challenge the dedicated troublemaker. Some encryption methods are easier to decode than others. Three basic techniques exist:

- Proprietary methods, which are not widely known but are also not verified as secure
- Data Encryption Standard (DES), a government standard that was unbreakable in its original design of 256-bit keys, and is still very secure with its present 57-bit keys
- Rivest-Shamir-Adleman Public Key (RSA), a slower but unbroken system that is not officially sanctioned

It may interest you to know that in 1981 Epic Computer Products demonstrated a prerelease version of an RSA public key

cryptosystem for personal computers. An agent of the National Security Administration contacted them. Epic's programmers explained that there was no way to insert a trap door to allow breaking someone's private code. Soon thereafter, Epic received a letter from the United States Bureau of Munitions and Firearms stating that the software could be neither exported nor sold to multinational companies. The product, *Kryptyk*, was never released.

Even if people can't read encrypted files, they can still damage them. First, they can remove a file entirely with DEL, ERASE, or more thorough utilities. Second, they can overwrite part or all of the contents with DEBUG or other utilities. Once damaged, a file cannot be decrypted reliably. *Hardware encryption*, discussed later in this chapter, prevents these kinds of roundabout damage.

Passwords

> Passwords are as valuable as data they protect.
>
> - Use long passwords of unrelated words and numbers.
> - Write them down in a safe place. Keep the written password under lock and key. Keep the key with you or in a secure key locker.
> - Change passwords periodically, but unpredictably.

Password protection is only as good as the password. Don't use your name, your initials, the application name, or anything obvious as your password. Don't leave the password where someone can find it.

Richard Feynman, Nobel prize winning physicist, had a hobby. He liked to pick locks. Combination locks were the easiest because he could almost always guess the combination. People used their birthdate, their wedding date, their age or something nearly as obvious. If he couldn't guess it, he could find it written on the corner of their blotter, inside their appointment book, or in their top desk drawer. Bear this in mind when dealing with your password.

Password protection can operate at many levels:

- To encode a file
- To control access to part or all of the data in a file
- To allow use of a program
- Within a shell or menu program, to allow access to particular applications or directories

Some application programs include password capabilities. Many don't. If security is important for you, put password protection on your shopping check lists.

Using Passwords in Menu/Shell Programs

Menu and shell programs were discussed in detail in Chapter 5. Even a simple batch file menu system can include passwords. However, it's easy to read the batch files and learn the passwords. Furthermore, you can always circumvent the system by starting from floppy disks or otherwise avoiding the batch file.

Most commercial shell/menu programs include at least one level of passwords. Many include more. For simple protection, organize the main menu by user: George, Mary, and so on. Assign a password for each user. Give each user a submenu listing all his/her applications. If you're the only user, make up one password for the entire system. More sophisticated shells distinguish between the system manager and ordinary users. The system manager has a special system password which gives access to all functions, including changing the password and menus.

Audit Trails

A few menu/shell programs keep a log, or audit trail, of file accesses. Every time someone uses a file, the program records who and when, and which file was accessed. Likewise, logs record unsuccessful attempts, when someone tries to use a file but doesn't know the password. Only the system manager can view the log.

Audit trails are a useful security feature. They are also handy for analyzing time spent:

- By different personnel
- On various projects
- Servicing different clients
- On business versus personal use (for the Internal Revenue Service)

Full-Blown Software Security Systems

If the features described so far still leave you feeling nervous, consider a comprehensive security system. Besides file encryption, passwords, and audit trails, software security systems offer

- Protection against booting from floppy disks
- Protection against modifying CONFIG.SYS or AUTOEXEC.BAT to circumvent the security system
- Lock-out protection to control access to any file, any directory, or MS-DOS
- Protection to allow only the system manager to copy files from listed directories (which will prevent piracy)

Watchdog is probably the most complete personal computer security package available at this writing. It's so thorough that you must clean out your entire disk and reinstall everything from scratch. You create password lists for every file or directory and every MS-DOS command. File encryption is automatic if you want it. The keyboard locks while booting, so no one can escape the *Watchdog* environment.

Watchdog goes beyond simple password protection: Within any one directory, users can

- Read
- Read and write
- Read old files only
- Create new files only

This is heavy-duty protection. It can be time-consuming and tedious to operate under its watchful eye. It's no environment for the sloppy user. You can lose all access to your system, or lose access to big chunks of it, if you don't keep good records.

Hardware Protection

Hardware protection includes locks and keys, removable hard disks, and sophisticated disk-protection expansion boards. It is easy for an expert to bypass many software security schemes to remove or alter a file. Most hardware protection is nearly impossible to bypass. Prices range from $30 to $3,000 and up.

Hardware locks. Start with what you already have: Does your office door lock? If so, lock it whenever you leave, even briefly.

The IBM PC AT and most compatibles come with a simple built-in hardware switch. You can retrofit similar locks to other personal computers. When the switch is off, the machine is useless. This is good basic protection. Unfortunately, hardware locks are

powerless against people with legitimate access to a machine and against anyone who can find the key.

Also, some locks are better than others. The IBM PC AT uses a special cylindrical lock. If you lose both keys, the company can make new ones if you know the lock number. Many clones use inferior locks, and there's no way to get a replacement key. One clone maker even used a fake lock, installing it only to make the machine look more like an AT.

Use more than one hardware lock for complete protection.

- Keyboard locks stop roving fingers even while the machine is in use, but they don't prevent hardware thieves.
- On-off locks stop usage but don't prevent stealing the entire system. Furthermore, technically-minded thieves can easily hot-wire the system and bypass the lock.
- Anchor bolts lock the machine to the desk or floor. No one will be able to walk off with the machine, but bolts won't protect your data.
- Backplate or coverscrew locks protect your machine's innards, hard disk and internal boards. Again, though, there's no data protection.

Hardware Locks: What They Can and Can't Prevent

	Stealing the Computer	Stealing the Hard Disk or Internal Boards	Unauthorized Use of System or Data
Locking keyboard covers	NO	NO	YES
On-off locks	NO	NO	YES
Anchor bolts	YES	NO	NO
Backplate/coverscrew locks	NO	YES	NO

Removable disks. Most removable hard disk units hold 10 to 20Mb of data apiece. Although costs have dropped greatly in the last few years, these systems still run $1,000 and up. Removable disks are appealing because they solve several problems at once because they

- Make backing up your files easier
- Protect data from outsiders because you can lock up the disks
- Make it easy to isolate different users' data by giving everyone his or her own disk

- Let you run different operating systems or environments easily
- Give you unlimited total storage

Removable disks don't solve all problems. You still need to deal with access to MS-DOS and other common functions. You also need to decide how to handle programs used by more than one person. If you have a nonremovable hard disk, put shared programs there. If not, put copies on each disk where needed. If copy protection interferes, you may have to run the program from floppy disks (see Chapter 9).

Security boards. Security boards equal and exceed the security provided by the most stringent software security systems. Hardware boards protect more thoroughly, more quickly, and with less possibility of tampering than with software systems. Unfortunately, hardware boards also cost more—$500 and up at this writing.

Hardware security features to look for:

- Data encryption. The encryption algorithm should be built into a chip. The chip itself should have protection so it won't work at all if someone tries to tamper with it.
- Password protection. Most boards keep password information in ROM. They check passwords during boot-up and will not release control for an invalid user. Passwords can restrict access on either the subdirectory or file level. Access can be of different types—read-write, read-only, and so on. Some boards can also restrict access to peripherals.
- Audit trails. Hardware boards usually store this information in their own RAM.
- By-pass protection. Hardware is generally more effective than software at keeping unauthorized users out of a system entirely or away from files that don't belong to them.

Protecting Shared Data

Everything discussed so far involves one machine and one user at a time. Networks, communications, and multitasking bring new headaches. A few network security devices are starting to appear, but most are still experimental. For now, consider:

- Is sensitive data encrypted when traveling across phone lines?
- What password protection/access restrictions protect your network?

Summary

In the security arena, as elsewhere, there are tradeoffs. Hardware security costs money. Software security costs time and convenience. File encryption, passwords, shells, and file manipulations all have pluses and minuses.

Study your environment carefully to see what kinds of protection make sense. To help you, here is a security worksheet. You may want to make one copy for each machine you need to protect.

System: ______________________________

Date: ______________________________

Done by: ______________________________

Security Management: Who's in Charge

Hardware security ______________________________

Maintaining the directory structure ______________________________

Maintaining menus/shells ______________________________

Coordinating access and passwords ______________________________

Checking/analyzing audit trails ______________________________

Hardware Security

What kinds of locks make sense for this system?

Locked room	____________
On-off key	____________
Keyboard lock	____________
Backplate lock	____________
Anchoring	____________

Data Security

Is the directory structure on this machine appropriate for your security needs?

____

What data on this machine should be protected?

____

____

____

____

What are the most appropriate ways to guard this data?

	Should do	Are doing
Offline storage		
floppy disks	______	______
Do we erase all online copies?		
removable hard disks	______	______
Other security techniques		
File encryption	______	______
Password protection		
Directory level	______	______
File level	______	______
Audit trails	______	______
Protection against booting without going through security	______	______
Complete lock-out protection	______	______
Protection against copying program files	______	______

Should we use hardware or software for the above? ______________________________

Accident protection

	Yes	No
Do we use menus/shells for inexperienced users?	____	____
Do we hide vulnerable files or make them read-only?	____	____
Do we keep FORMAT, COPY, and other hazardous commands safely out of reach?	____	____

Chapter 8

Backups

Backups

Backups are boring but crucial. This chapter covers the whys, whens, and hows of backup. You'll see how to select backup software, pick appropriate backup media, and develop a thorough backup plan.

> Have a complete, workable backup system and stick to it.
>
> - Backup often and regularly.
> - Use multiple backup methods.
> - Consider different media, locations, and frequencies for different backup jobs.

Why Backup?

Just to point out how easy it is to lose data, here is an anecdote just a few years old.

The public domain program *PC-Talk* is widely used for moving files to and from computer bulletin board systems (BBSs). Documentation for the early versions of the program was sketchy, but running *PC-Talk* was straightforward.

Transferring files was easy. You pressed Ctrl-T to transmit or Ctrl-R to receive, followed by the filename. To send similarly-named groups of files, you specified wildcard filenames using * or ?. Hapless users were dismayed to learn that to receive groups of files, you merely pressed Enter for the filename. Trying to receive *.* wiped out disk directories. Why? For efficiency, *PC-Talk* erased old versions of matching files to make room for the new incoming copies, so trying to receive a group of files designated *.* performed the equivalent of DEL *.*.

Backups protect against data loss when things go wrong. They guard against:

- Hardware malfunctions
  - Disk failures (usually occur after 8,000 to 20,000 hours of use)
  - Spindle, bearing, or motor problems
  - Shocks, bumps, or jiggles causing hardware damage
  - Bad sectors as a result of dirt, wear, head crashes
- Electrical/electronic malfunctions
  - Power surges
  - Brownouts
  - Electronic circuitry errors or failures
  - Accidental power-down
- Acts of God
  - Earthquakes
  - Fire
  - Floods
  - Hurricanes and tornadoes
- Theft
- Software quirks/bugs
- Accidents
  - Erasing a file by mistake
  - Copying the wrong way
  - Spilled liquids (the number one cause of computer damage)

Backup files are useful for normal processing.

- Spring cleaning: To unclutter your disk (but keep an offline copy just in case).
- Experimenting: Because you know you can always undo whatever damage you cause.
- Data interchange: To share data with someone else or to move data to another system.
- Installing software: Restore already-configured applications onto similary-equipped systems to avoid redoing the setup at every licensed site.

Different Types of Backup

By now, you're probably convinced that you must back up your files. So, how will you do those backups? You need to select appropriate backup software, which will be discussed in the next section. But first you must establish a backup routine. Here are the questions you must ask yourself:

- When?
- What?
- How many?
- Where?
- To what?

When: How Often to Backup

> Backup whenever you've done more work than you want to lose.

When writing software, save the program source code (text) to hard disk every few screens. At least once a day, copy the hard disk files to floppy disk. Also update the floppy backup every time you turn off the computer.

Good backup frequencies are:

- Every document page
- Prior to taking a break
- When you start work (or just before leaving)
- Weekly
- Monthly
- Quarterly
- Whenever you can't remember doing the last one

Decide whether you would rather start your daily backup fifteen minutes before quitting time, or ease into the day's activities with the task of backup. Once you pick your schedule, stick with it.

At least one product, Intellisoft's *Bookmark*, saves your work in progress. At intervals define by the user, it writes a snapshot image of your computer's memory and its internal registers to your hard disk. This $100 program can therefore backup your work to within

seconds. Fifteen-minute intervals are a good tradeoff for the overhead of backups and the effort to rekey.

Save My Day from Computer Foundations, Rochester NY, saves your keystrokes to disk as you type. You can automatically rerun your application when the need arises. This program does not handle software that directly reads the keyboard hardware, rather than using MS-DOS or ROM BIOS functions. Since most applications either avoid direct keyboard access, or provide an alternative installation for near-compatible systems, this should not be a problem.

Three factors determine backup frequency:

- Source availability
- Added value
- Backup cost

Source availability. How accessible is the original (source) material for this file? If the information was keyed from printed matter, did you retain the input documents or a photocopy of them? If the information was extracted or derived from other computer files, are those input files and the translation software still available? If you don't keep source materials, you need good, frequent backups.

Added value. How much has been invested in changing this file since its last backup? Consider all the resources involved, such as data entry time, interim processing to alter the file, and what reports and files have been affected. How volatile is your data?

Backup cost. What does it take to create the backup? This answer varies, as you have many options for making backups. Perform less costly backups frequently. Do major backups less often.

What: Incremental Versus Application Versus Wholesale Backups

You don't need to do a wholesale backup, copying your entire hard disk, just because one or two (or 20) files change. Trying to back up your entire hard disk every day is ill-advised. If you schedule 30 minutes to save your files daily, you will skip a backup occasionally. You're more likely to maintain a backup schedule if it only takes four minutes a day, plus 30 minutes once a week.

Backups have one of three scopes:

- Master backups
- Incremental backups
- Application backups

Master backups. Master backups include all your files or at least all data files. You can do these every month or quarter.

Incremental backups. Incremental backups copy everything changed since the last master backup. You can do an incremental backup on your entire disk or on specific directories or filenames. These are good candidates for daily use.

Application backups. Application backups copy a related set of files. For example, you might copy all accounting files. It's possible that not all the files have changed since the last backup. Nonetheless, copy them all to insure that they will work together as an integrated whole if you need them. Do this kind of backup after every major processing cycle.

Most backup programs can do incremental backups without any extra work by you. They use the *archive* attribute switch which MS-DOS maintains for every file. MS-DOS turns on the archive attribute whenever it creates or changes a file. Backup programs turn it off for each file they copy. Thus, an incremental backup simply copies all files having the archive attribute on.

How Many: Keeping Multiple Generations

Rules for *multigeneration backups:*

- Always keep at least two backup sets, preferably three.
- Rotate your backup sets systematically, overwriting the oldest with the new set.
- Destroy disks or tapes at the first sign of wear. Media is always cheaper than data.
- Be systematic. Test the completeness of your scheme.
- Verify that you are backing up good data, not propagating garbage.

Imagine that a very important file, one that took many hours to create, suffers catastrophic damage. You aren't aware that your beautifully crafted work has become random electrical noise. Prior

to presenting the results of your labor to the board of directors, you industriously copy the corrupted file onto your lone backup disk. After all, you have no reason to suspect there's anything wrong with the file. An interesting presentation is guaranteed.

An application consists of three classes of files:

- Masters
- Transactions
- Temporaries

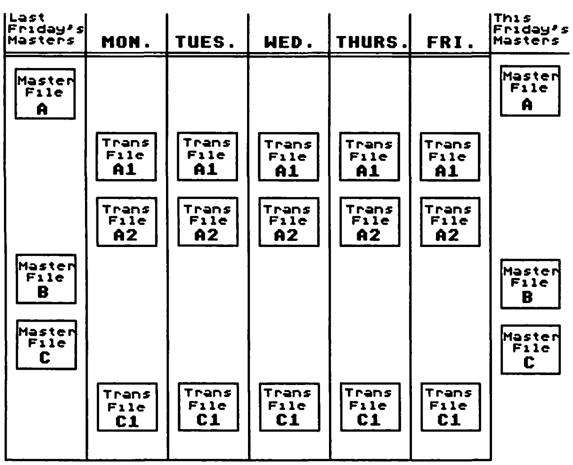
Masters. Masters contain semipermanent information, such as names, addresses, account balances, and stock on hand.

Transactions. Transactions contain short-lived data that affects Masters, such as time cards, vouchers, payments, and so on.

Temporaries. Temporaries are interim collections of data extracted from other files, such as reports, queries, or logs. Some interactive systems, such as online order-entry, generate copies of keyboard activity in transaction log files. Other interactive systems, such as word processors, directly affect the master file.

A single generation of backup consists of all the master files and subsequent transactions necessary to regenerate the next version of masters, as depicted in Figure 8-1. Temporary files are not retained, unless regenerating them would be prohibitive.

Figure 8-1. A single backup generation consists of the master files and transaction data necessary to create the next generation's master files.



Test the completeness of your backup generation.

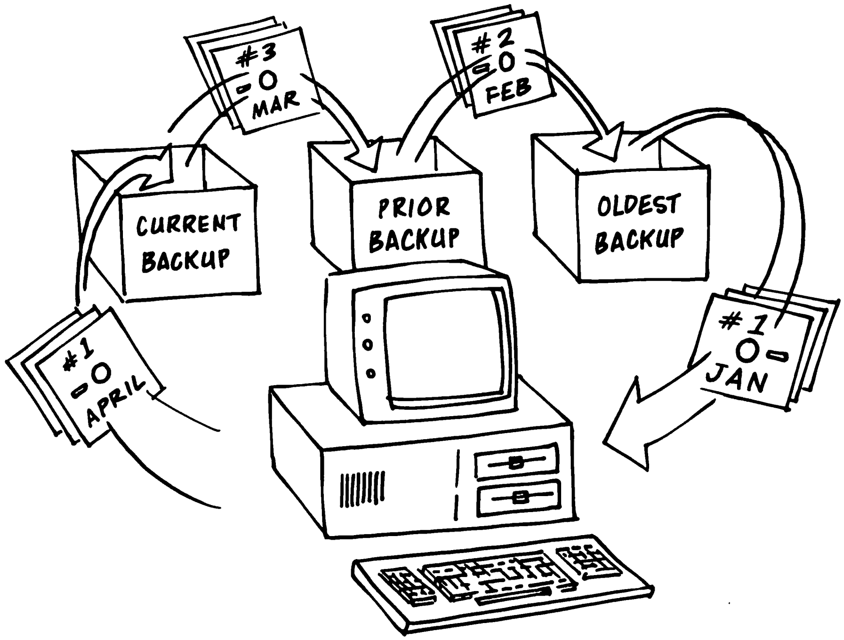
- Back up up the entire application, file-by-file if possible. This is your *Start Set*.
- At the end of the next complete processing cycle, make another file-by-file backup of the entire application. This is your *End Set*.
- Erase all the application's data files from the hard disk.
- From your *Start Set*, copy those files you have identified as the backup generation onto the hard disk.
- Run whatever programs are necessary to generate the new master files.
- As you discover the need for additional files, add them to your backup generation set and copy them onto the hard disk from your Start Set.
- When you have created new master files, run a binary file comparison program to check them against the End Set files.
- If you had to add even one file to your backup set, or if you had to rerun any programs during the creation of the duplicate master files, erase the hard disk files, copy the backup generation files, and recreate the masters again. Make sure that the backup creates your masters without any problems.
- Write a batch file (see Appendix B) that copies the backup generation files from the hard disk to the backup media. This is your *Application Backup* job.
- Write a complementary batch file that erases the hard disk files, copies the backup generation set onto the hard disk, and runs the software to recreate the master files. This is your *Application Restore* job.
- Include the Application Backup and Application Restore batch files in your backup generation set.

In a multigeneration backup, the oldest generation is constantly replaced by the newest. A three-generation backup lets you go back at least two full processing cycles. Figure 8-2 summarizes a simple three-generation monthly backup.

The key to multigenerational backup is being systematic. Here is a description of one system that works. Assume that every month-end you copy all of your cost-accounting files onto four floppy disks. You also make a weekly data entry backup, four per month.

You need 24 floppy disks—three sets, each containing four master disks and four weekly update disks. Use three boxes to hold

Figure 8-2. The new backup replaces the oldest generation.



your three sets. Label the boxes *Current Cost Accounting Backups, Prior Cost Accounting Backups,* and *Oldest Cost Accounting Backups.* In the Current box, put a sheet of paper with two headings: *Backup Set* and *Backup Date.* Label each disk with its disk number and set number, from *Set 1 Master Disk 1,* to *Set 3 Weekly Disk 4.*

After the month-end processing, take the eight disks out of the Oldest box, move the Prior disks to the Oldest box and the Current disks to the Prior box. Reuse the Oldest Master disks in sequence (Master Disk 1, then Master Disk 2, then 3, and so on) to make your new Master backup set. Refer to Figure 8-2.

Put the entire set in the Current box. Enter the set number and the date on the sheet of paper in the Current box. Each week, copy the transaction backup to the proper Weekly disk within the Current box and update the log sheet.

Now, if your Current set is damaged in any way, you have another complete backup set, one week Prior. Even if you also lose the Prior set, you have a third Oldest set. If you ever need to go back and undo a mistake, you have three entire processing cycles in which to discover and correct it.

Where: On-Site Backup Versus Off-Site

> Keep at least one complete set of backups in a different building far from your computer.

You have your three sets of backups, neatly labeled and systematically rotated. If they sit next to your computer in three boxes, they will be neatly, systematically lost in the same catastrophe that destroys your computer, as happened to a small computer-repair business located in an older light industrial park: They were pleased when the landlord began to renovate. The parking lot had been resurfaced and the building was repainted. Reroofing fixed slight leaks and skylight installation had begun.

Eager to begin their Labor Day weekend, the reroofing crew neglected to secure a tarpaulin over the partially-removed roof. So, of course, the weekend was unexpectedly rainy.

The insurance company replaced the repair firm's equipment as well as their customers' equipment. But the company still nearly went out of business because all their computer files—including the floppy disk backups—had been destroyed by water damage.

Off-site backup doesn't have to mean a fancy or expensive vault (unless the value of your data justifies the expense). If your computer is at work, keep your Prior backups at home. If your computer is at home, keep the Priors at work, at a friend's house, or in a safe deposit box.

To What: Backup Media

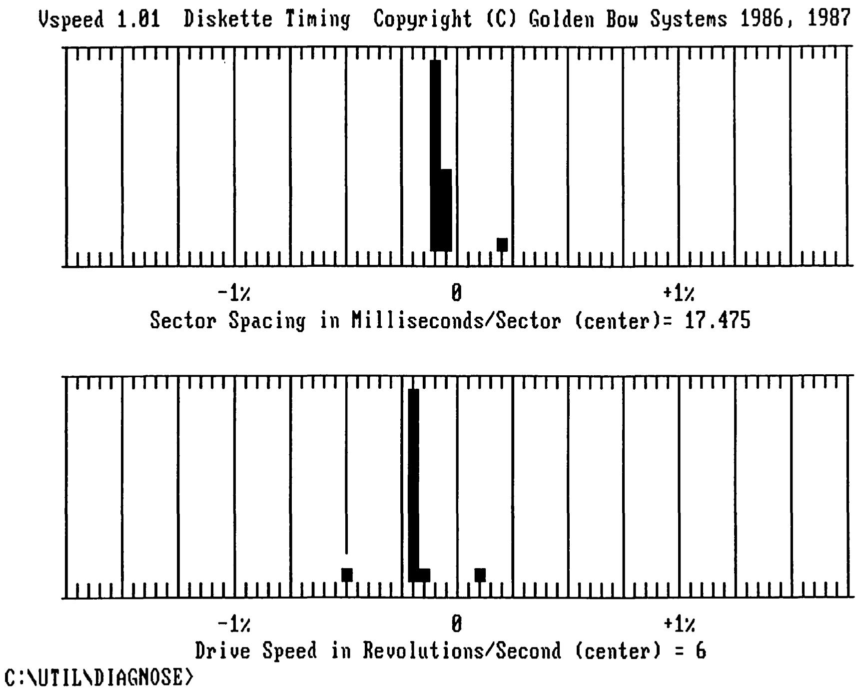
Most people think of floppy disks when they think of backups. Floppy disks are by far the most common backup medium. However, other backup media exist and may be more appropriate for at least some of your backup needs.

Even if you do not rely solely on disks for your backup medium, you will probably be making disk copies of files as secondary backups. You need to keep your disk drives adjusted close to specifications. If system A's drive is too slow, disks will have larger sectors. If system B's drive is too fast, the aggregate speed difference could approach 1.5 percent, and B will not be able to read disks from A. If system A's drive continues to slow down gradually,

last July's monthly backup could be unreadable when you really need it for consolidation reporting in December.

Various disk diagnostic software aids in determining if your drives need service. VSPEED, part of the *Vopt* package from Golden Bow Systems, measures the rotational speed of disk drives and the sector spacing of disks. Figure 8-3 shows the analysis of a drive that successfully reads a disk which was formatted upon another drive. Minor adjustments in drive speed can result in error-free disk data interchange.

Figure 8-3. VSPEED measures the rotational speed of the drive and the sector spacing of the disk.



Other media for backup include:

- Redundant disks
- Removable hard disks
- Quasi-floppy media

- Tape
  - Half-inch open reels
  - DC6000 tape cartridges
  - DC2000 tape cartridges
  - DC1000 tape systems
  - VCR
- Optical Devices
- Hard Copy

Redundant disks. A few manufacturers offer dual disk systems. For instance, the Shadow from Ford/Higgins holds 86Mb of formatted data on each drive. Each disk has its own controller, power supply, fuse, and power switch. Data goes simultaneously to each device. Support software tells you of problems with either drive. Housed in an external unit, the Shadow weighs 24 pounds and can be removed and locked away safely at night.

These systems greatly reduce the need for other backups. However, many factors could dictate additional backup. Off-site storage and archival storage requirements beyond the disk's capacity are two examples.

Removable hard disks. Removable hard disks can be internal or external, holding 10Mb to 100Mb each. Removable disk systems cost more than nonremovable systems. They are also usually slower, have greater danger of environmental damage, and less resistance to shock damage. They are often smaller than fixed disks because they contain only a single platter.

Yet, combining hard disk speed and storage capacity with removability yields virtually unlimited hard disk storage. Unlike the dual disk systems described above, removable hard disk cartridges can handle all your backup needs, including off-site and archival storage.

Matched removable hard disk drives are even better. Then backup becomes a simple task of copying an entire disk from one drive to another.

Quasi-floppy media. Superficially, the Bernoulli Box looks like an external removable hard disk unit. It's more a hybrid of a floppy medium with a hard disk read-write head. Floppy disk media is encased in a rigid cartidge, which inserts into the front of the drive. A unique air flow technology keeps the normally flexible disk rigid while it spins past the read-write head.

The disk moves up to the recording head. Any shock or jolt flops it down, away from the head. As a result, data is extremely safe from environmental or shock damage. Each cartridge holds 10Mb or 20Mb. Each unit contains either one or two cartridge drives.

Kodak has developed a 12Mb cartridge system using traditional floppy technology. Finer media and more accurate motor and heads allow the greatly increased capacity. The Kodak system is as slow as the slowest hard disk systems. However, its cost, about half that of a Bernoulli Box, makes it a contender.

In hard disk systems there is no contact between the head and the recording surface. Indeed, any such contact causes media damage—a *head crash*. There is no media wear in hard disk systems. Conversely, quasi-floppy cartridges suffer from surface wear and grit accumulation. Like floppy disks, the recording media touches the drive head during operation. Friction rubs the magnetic oxides off the disk, causing them to accumulate on the head.

Tape. Tape units can be internal or external, using your computer's power supply or their own. At least four different tape standards exist, using eighth-, quarter-, or half-inch tape. Tapes typically hold between 10Mb and 60Mb. Prices start around $800 retail, lower than removable disk drives but higher than a second internal hard disk.

Tape media costs less than disk media—as low as $.50 per megabyte for tape versus $5 or so for removable disks. Tape units are usually slower than hard disks but faster than floppy disks. Quarter-inch streaming tape is the medium of choice in large companies. More than one million units were shipped in 1986.

Here's a brief look at the major types of tape systems available.

Half-inch open reels. The traditional backup medium for minicomputers and mainframe computers is nine-track, half-inch open-reel tape. Half-inch tape units are available for microcomputers, too. They are far larger and costlier than other micro-based tape units—the reel alone is ten inches across and an inch thick. These drives cost from $3,500 to $10,000.

However, half-inch reels have two big advantages. First, because the medium is so roomy, the data is sparse and therefore unlikely to degrade in storage. Second and most important, nine-track

half-inch tape is the only consistent interchangeable data format. Your half-inch tape can move, unchanged, to dozens of other machines of all sizes and manufacturers. If you know you'll be swapping data with other systems, consider half-inch tape.

DC6000 tape cartridges. A DC6000 cartridge is a high-density quarter-inch magnetic oxide tape mounted on a rigid aluminum baseplate, protected by clear plastic with a hinged door. The technology was developed by 3M, who licenses it to other manufacturers. A special friction band moves the tape without ever touching it.

Like half-inch tape, DC6000 cartridges use nine tracks. However, traditional tapes record data in parallel across all nine tracks. Cartridge units record data in long strips, filling one track completely before starting the next. A servomechanism moves the head the right distance from the tape edge.

Once small and slow, DC6000's today typically contain 60Mb or 120Mb and stream at 90 inches per second, yielding transfer rates of 3MHz to 5MHz—as fast as most hard disks. Drives cost $600 to $1,500. Standards did not evolve as fast as the machines themselves, so there is very little compatability among different manufacturers.

Size can be a problem with DC6000 units. Each cartridge is about as big as a paperback book, and internal drives may use one-and-a-half or even two slots in your machine.

DC2000 tape cartridges. 3M down-sized the basic DC6000 design to create the DC2000. These cartridges are about the size of an audio cassette. By using 12–24 tracks and a higher recording density, they typically squeeze in 40Mb of data per cassette. Fully one-third of the space is used for formatting and error correction data. As a result, the predicted error rate for these tapes is lower than for most disk drives. DC2000 drives range in price from $600 to $1,200.

Irwin Magnetics placed servo signals on the tape to boost DC2000 capacity to 64Mb. Their Model 465 is an external unit and costs $1,100. It reads 10MB, 20Mb, and 40Mb tapes from earlier Irwin drives.

Two well-defined standards, both developed by the Quarter-Inch Compatability (QIC) committee, dominate the DC2000 world. QIC-40 is geared for PCs. It defines the low-level recording format and the tape speeds. Since QIC-40 tape speeds are designed to

match floppy disk transfer rates, these units are slow. QIC-100 allows manufacturers to develop their own controllers. It's a looser standard and typically yields faster transfer rates.

Unfortunately, neither standard defines file structures. As a result, although the QIC DC2000 standards are far more universal than after-the-fact standards for DC6000 systems, compatability is still lacking. Low-level formatting compatability is not enough: Different units can read and understand one another's data character by character, but they don't know how it fits together in files.

DC1000 tape systems. Just as compact, just as slow, but less expensive than DC2000 units are DC1000 tape cartridge systems. Sharing the same 3M technology as DC6000s and DC2000s, DC1000 systems use .15 inch tape, commonly called quarter-inch tape. Sold by several manufacturers but dominated by Irwin Magnetics, DC1000 systems are a popular choice because they cost as little as $300. One disadvantage is their small capacity, 10Mb to 20Mb.

Tape software. Almost all modern tape units come with backup software. Ideally, a unit should be able to perform three kinds of backup:

- Image backup
- Complete file-by-file
- Partial file-by-file

Image backup is a character-by-character copy of the entire hard disk. This is the fastest type of backup, and the fastest for restoring an entire disk. However, it is nearly useless for restoring an individual file or group of files. After all, how are you going to locate one file within that mass of characters? What's worse, a single file may be in several pieces on the disk, making it even more difficult to reconstruct.

A complete file-by-file backup is another form of total backup. It identifies and copies files separately. The program keeps track of both files and directories. Unlike an image backup, a file-by-file backup can provide single files for restoration, although slowly. Unless your disk is less than two-thirds full, total file-by-file backup probably takes longer than an image backup. However, the ability to restore single files is worth the additional time for most people.

A partial file-by-file backup works just as with any other backup medium. You should be able to backup and restore selected files, groups of files, or directories. Once rare, selective backup is now available on most tape units.

Sytron, of Marlboro MA, offers SY-TOS software for QIC-40 DC2000 tape drives. Costing $100, it offers the same backup options as for the IBM 6157 tape unit. This software allows backup in file-by-file, selection-by-date, file-name, image-streaming, and other formats. It can restore selected files from an image backup. It also supports password protection.

VCR. VCRs are a recent low-cost replacement for tape backup units. They require a special controller but use an ordinary VCR. Controller prices start around $350 retail. Controllers typically use one expansion slot and include software for all three backup types.

VCR backups utilize ordinary, inexpensive VCR tape cassettes. Each tape holds between 50Mb and 150Mb. VCR controllers ensure data integrity by making four to 40 copies of each file.

Unfortunately, VCR backup is slow, often slower than the slowest tape units. Furthermore, keeping multiple copies of each file slows restoration of a single file or group of files.

Nonetheless, VCRs are an attractive, low-cost alternative to copying onto dozens of floppy disks.

Optical devices. Optical storage is the newest family of high-tech wizardry. Traditional computer storage media are magnetically based: Bits of information are recorded as positive or negative magnetic states. Magnetic media are flexible and ever-improving. However, magnetic media have two inherent problems:

- It is extremely difficult to pack data beyond a certain density.
- Outside magnetic fields easily affect the magnetic storage medium and damage data integrity. Since magnetic fields are so common in the environment, data degradation is a frequent and serious problem.

Optical media overcome both problems. Optical technology records data bits as either of two different reflective (optical) states. Rather than being polar opposites, like magnetic data states, the two optical states differ by only a few percent.

A laser reads and/or writes the data. Laser technology is highly accurate by nature, so extremely tight data packing is possible. An optical drive typically holds 400 times as much data as a similarly-sized magnetic device.

Three kinds of optical storage are available:

- CD-ROM (Compact Disk-Read Only Memory)
- WORM (Write-Once-Read-Many)
- EOD (Erasable Optical Disks)

CD-ROM. CD-ROMs are read-only devices similar to audio compact disks. They are ideal for high-volume data dispersal. Encyclopedias on disk are a perfect example. Because the high-energy data recording takes place at a central location, and because the duplication and read-only technology is so similar to that already available for audio systems, costs are relatively low—$1,000 and up versus $3,000 or more for other optical technologies.

WORM. WORM drives are the next step up in flexibility. These drives include a high-energy laser for data storage. When writing data, the laser actually creates tiny permanent holes in the recording surface. It's easy to understand why you can only write once.

Permanence can be a disadvantage or an advantage. For environments requiring large amounts of archival storage, WORM drives are perfect. You can store huge amounts of data in a small space. Data degradation is minor. The life expectancy of data on WORM storage is around ten years, compared to three for the best magnetic media.

EOD. If optical storage is the new family on the block, EODs (Erasable Optical Disks) are the new family's new baby. WORMs do their job reliably, but what if you don't want permanence? The high capacity of optical technology has led a number of manufacturers to work on reusable optical disks. These generally use a combination of magnetic and optical effects to change reflectivity states in an optical medium. The basic technology has been available for years, but creating workable, cost-effective systems is proving extremely difficult.

Practically speaking, only WORM drives currently offer optical backup support. At least six manufacturers offer WORM drives at prices from $3,500 to $10,000. All use one of two optical cartridge drives, from Optotech or Information Storage. The drives are the

size of full-height floppy drives. Both can use either single- or double-sided cartridges, although you must manually flip the cartridges to use the second side. Optotech cartridges hold 200Mb per side; Information Storage's hold 120Mb per side, with cartridges holding 500Mb per side coming in the near future. Double-sided cartridges cost about $225 apiece and single-sided are $150.

Hard copy. Often people falsely assume backups must be machine readable. Backup includes anything enabling recovery from data damage. Since printed listings are an important part of computer output, why not use them?

No one wants to manually recopy data from listings, but it's better than nothing. If regularly printed reports contain sufficient detail to reconstruct files, you may be able to make backups less often. An occasional printout of your directory tree and batch files helps both backup and documentation.

Not all hard copy backup must be rekeyed in case of catastrophe. Three methods exist for machine-reading hard copy:

- Optical character recognition
- Bar-code scanning
- Proprietary encoding

Optical character recognition. For as little as $90, you can purchase a hand-held character scanner. These devices are able to read a few typefaces and print sizes. They do not read dot-matrix printouts. As they decipher text, OCR units send ASCII characters to the computer.

Bar-code scanning. Bar code wands, the size of a thick pencil, are available for about $100. Most are driven by special software within the computer. At least one unit, the size of a credit card, reads and stores data internally. It is later connected to the computer for down-loading.

Proprietary encoding. Cauzin Systems offers their SoftStrip scanner for $125. This device reads data printed in strips, encoded into a proprietary format. Strips can be printed (slowly), using the graphics mode, on an Epson or IBM dot-matrix printer. Their *Archivist* package cleanly and easily prints strips with a laser printer. A single 8½ by 11–inch page can hold nearly 8K of binary data, complete with error-correction encoding. SoftStrips are well suited to archiving, such as keeping correspondence files off your hard disk but easily retrievable.

How: Backup Programs

You have two main backup software routes: MS-DOS or non-DOS.

MS-DOS. Three MS-DOS backup options are available for you right now:

- COPY
- BACKUP/RESTORE
- XCOPY

COPY. COPY is the easiest, most straightforward way to duplicate a single file or a related group of files. It's a built-in MS-DOS command, available any time you're in MS-DOS. It has some serious limitations, so you may not want to use it.

- It can't copy a file larger than the destination medium. For example, most floppy disks hold about 360K (360,000 characters). COPY can't copy a 400K file to a 360K floppy.
- COPY can't traverse directory trees. It copies all files in a directory, but it won't copy the files within any subdirectories.

The format of the COPY command is:

COPY ***sourcefile destinationfile*** **[/V]**

The *sourcefile* and *destinationfile* names can include paths and wildcards. If you give a path but not an actual filename for the destination, COPY gives each destination file the same name as its source file. For example, the following command copies all files with the .TXT suffix from directory \WP\DATA of drive C:

COPY C:\WP\DATA*.TXT A:

The copies will go onto a floppy disk in drive A and will have the same names as their original counterparts on drive C.

When COPY writes the destination file, it replaces any existing file in the destination directory that has the same filename. It doesn't ask you first. This can be dangerous. But you can use this to your advantage to *time-stamp* files.

Whenever MS-DOS creates a new file, it saves the current date and time as the time stamp of the new file's directory entry. A duplicate file created via COPY inherits the creation date and time of the original file. However, when COPY creates a new file, such as by combining multiple files (*appending*), it assigns a new time

stamp. If you tell COPY to append nothing to a file, it merely updates the time stamp of the original file. As in Figure 8-4, keep track of when you did your last backup within the time stamp of a small text file, BACKEDUP.ON.

Figure 8-4. Use COPY to time-stamp a file by appending nothing.

```
D:\DISKBOOK>type backedup.on
this file's time-stamp indicates the last backup

D:\DISKBOOK>dir backedup.on

 Volume in drive D has no label
 Directory of  D:\DISKBOOK

BACKEDUP ON        51   7-01-87   8:03a
        1 File(s)   8431616 bytes free

D:\DISKBOOK>copy backedup.on+,,
BACKEDUP.ON
        1 File(s) copied

D:\DISKBOOK>dir backedup.on

 Volume in drive D has no label
 Directory of  D:\DISKBOOK

BACKEDUP ON        51   8-30-87   6:11p
        1 File(s)   8431616 bytes free

D:\DISKBOOK>
```

The COPY /V option verifies that your copy is readable. COPY with /V takes nearly twice as long as without, because MS-DOS reads each file it writes. Using /V doesn't give 100 percent reassurance. /V says your new file is readable, but doesn't guarantee it's identical to the original file. Binary file comparison programs do, however.

BACKUP/RESTORE. COPY cannot handle files larger than the target medium, nor can it traverse subdirectories. MS-DOS BACKUP/RESTORE can do both, but their use is not recommended.

Several versions of MS-DOS (and PC-DOS) have bugs in BACKUP. Versions 3.0 and 3.1 are the worst. MS-DOS 3.0 BACKUP frequently omits a critical control file, without which RESTORE won't work. If you have MS-DOS 3.0 or 3.1, and you insist upon using BACKUP and RESTORE, get corrected programs from your

dealer. With version 2.*x* MS-DOS, proceed with caution. Do some trial runs before making BACKUP/RESTORE your standard backup method.

BACKUP files are unreadable except by the RESTORE program. This can be very frightening. If RESTORE can't read your BACKUP files, nothing can. You have to trust that BACKUP creates a good image, because you cannot use a file comparison program to verify the accuracy of your backup.

BACKUP and RESTORE are also considered too slow to be useful. Perhaps you should look into the purchase of a commercial backup/restore facility instead of these MS-DOS programs. Assuming that you followed the suggestions in Chapters 4 and 5, insert BACKUP.BAT and RESTORE.BAT into your \BATCH directory. The standard program search PATH will use these job streams to invoke your commercial utility.

XCOPY. Both COPY and BACKUP/RESTORE have major weaknesses. Fortunately, MS-DOS versions 3.2 and later offer a third choice. XCOPY combines and improves both COPY and BACKUP. Its command format is

drive \path \XCOPY source destination [/A /D:date /E /M /P /S /V /W]

where *destination* is either a filename, as in COPY, or just a drive, as in BACKUP/RESTORE.

Many XCOPY options work just as in BACKUP.

- /D:date copies only files later than a given date.
- /M copies only changed files (files with the archive attribute).
- /S copies files from subdirectories as well as from their root directory.
- /V turns on Verify mode, just as in COPY.

Three new options give added flexibility.

- /A copies files with the archive bit on, just like /M, but doesn't turn the bit off afterward.
- /P pauses to ask about each file it copies, before copying.
- /W waits while you insert any needed source disks. Thus, you can XCOPY from multiple disks.

Non-DOS. The MS-DOS backup commands discussed above can fill all your backup software needs, if you don't keep a large volume of data files on your hard disk and you're willing to invest time in backups. Sooner or later, most people end up with too many files and too little time to make these DOS commands practicable. Either they do backups less and less often, creating more and more risk, or they get a non-DOS backup program.

A large secondary market of non-DOS backup software exists. At least a dozen programs are available. What follows is a look at what to consider when selecting a program:

- Reliability
- Cost
- Speed
- Flexibility
- Ease of use

Reliability. You'd think any stand-alone backup program would be reliable, regardless of other features. Unfortunately, this isn't true. Many stand-alone programs fail, frequently when reading their own backup files. Study carefully before buying backup software: Read computer periodicals; ask friends; and assume nothing. Find out how long this program has been available and how many are in use. What version are they offering? If it's the first release, be careful.

Some programs use standard DOS-readable files. Some use their own formats that only they can read. If it's a special format, DOS-compatible file recovery programs such as *Mace* and *Norton Utilities* won't work. Do you feel comfortable with this? Does the program's reliability warrant it?

Almost all backup programs say they verify readability. This means they check to make sure they can read the file they've just written. An extra level of safety comes from a character-by-character comparison of the original and the copy. Check whether it's available, and whether it's automatic.

Finally, find out how the program handles disk errors. When it finds bad sectors, it should mark them as bad, avoid using them, and keep going. Most do. A few don't. They give the MS-DOS prompt *Abort, Retry, or Ignore?* message, or they die. You're stuck restarting your backup from the beginning. Besides marking the

bad sector and plowing on, a good program should warn you if a disk contains a large numbers of bad sectors. You probably don't want to use such a marginal disk for your precious backups.

The safest programs go one step further to help you recover from data errors.

Speed. If a backup program is no faster than MS-DOS BACKUP, why bother? Most commercial backup programs claim a 200 to 300 percent speed advantage over MS-DOS. It's not necessarily true.

Commercial programs gain speed several ways. They generally control the hardware directly, rather than going through MS-DOS. They can thus optimize transfer operations. For example, MS-DOS uses only one Direct Memory Access (DMA) channel for transfers, even though two are available. Many backup programs reap much of their speed advantage simply from using both channels where possible.

At least one program, *FASTBACK,* gains speed by keeping the floppy drive spinning all the time. It's frightening to ignore all your training and open the drive door with the red light on, an action required while using this software. Nonetheless, you'll gain much time, without danger, by keeping the drive at full speed.

Commercial programs are more organized than the MS-DOS commands when writing to floppy disks. Writing a file involves several steps:

- Check the file allocation table (FAT) for space.
- Update the FAT.
- Write the file.

MS-DOS cycles through each step sequentially for each file it writes. The read-write head moves constantly between the FAT area and the file area, slowing the backup each time it moves.

Smart backup programs accumulate a full floppy disk's worth of file information, check and write the FAT entries at once, then write the actual files in smooth, back-to-back sequence. Thus they minimize head movement and backup time.

Many backup programs can format disks much faster than MS-DOS. The fastest take only about 20 seconds per disk.

Cost. Backup program prices range from free (public domain

offerings) to nearly $200. It's not always true that you get what you pay for in the backup arena. Choose the features you want, then worry about cost.

Flexibility. Good backup programs give you many choices, including:

- Image backup versus file-by-file backup
- Command line versus menu operation
- Backup catalogs or on-the-spot selections
- Multiple file names
- Selection criteria
- Attended, unattended, or background mode
- Choice of backup media

Image backups, discussed earlier in this chapter, are fast. File-by-file backups are better suited for partial restorations.

Usually menu-driven programs are easier to learn and use than command-driven programs; there are exceptions. Command line operation, where you add operating selections to the program name on the MS-DOS command line, is faster for power users and better suited for batch file job streams.

If you have a good backup plan, you'll be repeating the same backup jobs regularly. Good backup programs let you define tasks once for use when needed. A catalog is a list of files that are included in the backup. With catalogs, you can have a different set of backup files for each user or application.

The MS-DOS backup commands can only name one file or family of files to copy. Better backup programs allow you to use several names. The best have no limit.

A healthy backup plan includes both incremental and wholesale backup. Therefore, you should select files for backup based upon various criteria:

- All files in a directory
- Only changed files (archive bit on)
- Files dated before or after a certain date
- Include/exclude files within subdirectories

All backup programs run in *attended mode*. This means that you manually start and control the process. In addition, some programs offer *unattended* mode, in which you tell it to start copying

at a future time. Finally, a few programs offer *background* mode. Backup programs in background mode copy any changed files whenever the computer is idle. Both unattended and background modes require tape or hard disk backup for full benefit.

Some programs only write to floppy disks. Some write to any medium. If a tape drive or a second hard disk may be in your future, plan ahead.

Ease of use. User-friendly programs include an install procedure. The installation copies all program files, possibly onto a new subdirectory. Then it walks you through the steps necessary to tailor the package to your hardware and other needs.

The software you choose should have a complete, readable manual with an index. The manual should describe error handling as well as normal operating conditions. The best manuals also include strategy tips. One program even includes disk labels to help put your backup system into action.

Even more important, find out how often you'll need the manual. A good program should be nearly self-explanatory. It should have online help which changes as you move through the program.

Check whether there's a guarantee and online support. Find out about updates. How often will they be issued? Is there a charge?

Some programs use standard MS-DOS formatted disks, some use their own formats. In either case, find out if the program formats disks for you and if you can format them ahead of time.

Most programs cause the computer to beep when it's time to switch disks. A few don't. By your twentieth disk, you will appreciate not having to watch the drive.

The program should create a log file telling you exactly what was done in the backup.

Some Commercial Backup Programs

We have looked at the strategy for selecting commercial backup software. Now we will examine some specific products.

PMOVE. PMOVE is one of three file-handling shareware programs from Norm Patriquin. Its mind-boggling flexibility surpasses even the most expensive commercial program. Here are a few examples of its possible switch settings:

- /D lets you copy files older than, younger than, or equal to a given date.

- /E copies only files that exist on both the source and destination. This is an elegant way to copy only files you want.
- /F and /L (First and Last) copy files starting or ending with the given filename, based on directory sequence. In other words, /F:file.txt copies all files after reaching file.txt in the directory.
- /M merges two directories. The result is a single directory containing the latest version of all files from either directory. The source directory ends up empty.
- /SA saves duplicate files. If a duplicate exists on the target directory, *PMOVE* changes the last two characters of the extension.
- /T (Test mode) lets you "try before you buy." It shows what the results of a particular backup would be without moving any files.

An example of the sophistication of *PMOVE* is the way it handles moving files from one directory to another on the same disk volume. Rather than making a duplicate file and then deleting the original, *PMOVE* merely alters entries within the source and destination directory blocks. If it needs to create a target directory, *PMOVE* will ask for permission, as in Figure 8-5. Use *PMOVE* when you reorganize your directories.

Figure 8-5. *PMOVE* changes directory blocks to move files within the same disk volume, creating a target directory if necessary.

```
C:\>pmove c:\mace\256k c:\util\mace\256k
Directory c:\util\mace\256k\ does not exist.  Create it? (Y/N) Y
Directory c:\util\mace\256k\ created.
--- Moving from: c:\mace\256k\*.*
--- Moving to:   c:\util\mace\256k\
#  1 --  Moved with rename -- NEW    10240  MACE.EXE
#  2 --  Moved with rename -- NEW    36370  RX.EXE
#  3 --  Moved with rename -- NEW    12450  RXINIT.EXE

Selected 3 files containing 59060 characters
```

Fastback. Fifth Generation Systems established the category of commercial PC backup software with their *Fastback* product. The slogan they used that jarred the microcomputing world was *10Mb in 10 minutes. Fastback* programmers designed their program to use the direct memory access (DMA) channels of the disk controller integrated circuit so it transferred data between the hard disk and the floppy disks simultaneously, almost as a by-product of having MS-DOS access the hard disk data. *Fastback* created a marketplace in which blinding speed was a minimum requirement.

Fifth Generation Systems removed copy protection and added robust error detection and correction for *Fastback* version 5.13, which retails for about $125. However, it's harder to use and less flexible than newer programs. One handy feature is the ability to control, on a file-by-file basis, whether the Restore option overwrites old files, as shown in Figure 8-6.

Figure 8-6. The *Fastback* Restore will not overwrite old files without permission, either file-by-file via <R> or for all files via <Ctrl-R>.

```
Fifth Generation Systems, Inc                                    v5.13
7942 Picardy Ave., Suite B350    F A S T B A C K (tm)      (c) 1984, 1985
Baton Rouge, LA  504-767-0075                         all rights reserved

 EXISTING FILES ON DRIVE C: WILL BE REPLACED AFTER OPERATOR CONFIRMATION

     ↑  prev file            S  search for file      Esc  leave program
     ↓  next file       Ctrl-S  search again      Ctrl-C  stop (abort)
 Pg-Up  prev directory       V  verify file        1,2,3  'replace' options
 Pg-Dn  next directory  Ctrl-V  verify files           +  use catalog dir
                             R  restore file           *  select hard disk
                        Ctrl-R  restore files
                                                  Disk #1      17-Jun-87 01:26
 A: → C:     0 sectors reconstructed   File starts on #1      17-Jun-87 01:26

   20k  29-Jan-86 18:35  \REFLEX\RFLXINST.COM
   18k  21-Feb-86 00:00  \REFLEX\README.COE
  262k  21-Feb-86 00:01  \REFLEX\REFLEX2.EXE
  268k  21-Feb-86 00:00  LREFLEX\REFLDX.EXE
   10k  21-Feb-86 00:00  \REFLEX\READ-ME.RFX
    4k  27-Feb-86 10:52  \REFLEX\DRIVER.RX
  168k  28-Feb-86 01:10  \REFLEX\HELP.RXH
→   2k  29-Jan-86 15:22  \REFLEX\PC3270.RXR
```

> Be careful if you run *Fastback* on a machine with two disk drives. To achieve maximum speed, Fifth Generation programmers use both floppy disk drives for backup and restore. They expect your first restore disk to be in drive A. For a backup, they presume the disk in drive B is the first disk and reformat it *without any warning*. Many a user has had the disk in drive B *unintentionally* overwritten by *Fastback*. Create a batch file that reminds you to remove any disk from drive B. On a single-disk system, *Fastback* waits for you to verify that it can proceed.

COREfast. *COREfast* is fast, easy to use, reliable, reasonably-priced (about $150), and extremely flexible. It provides onscreen menus with context-sensitive help, as can be seen in Figure 8-7. It also runs from the command-line, making it well-suited to batch job streams (see Chapter 5).

Figure 8-7. *COREfast* provides image or file-by-file backup/restore with menus, help, speed, features, flexibility, and reliability.

```
COREfast          Version 1.33                 Monday, August 31, 1987

                        Backup Options Screen

       Backup type selected: File-by-file
    f2 - File specification: C:\UTIL\COREFAST\*.*
    f3 - Backup diskette drive: A:         f8 - Include subdirectories: Yes
    f4 - Volume name: UNNAMED              f9 - Only modified files: No
    f5 - Backup mode: High-speed         s-f1 - Read verify: Yes
    f6 - Diskette capacity: 1.3MB (HD) s-f2 - Use error correction: Yes
    f7 - Date range: All                 s-f3 - Update archive status: Yes

    f1  - Help
    f10 - Begin backup with selected options

Press ESC to exit

          (C) Copyright CORE International, Inc.  1985, 1986, 1987
```

COREfast performs an image backup of an entire device, or it makes a file-by-file backup under the direction of a *catalog* file. The catalog file is an ASCII file which lists the drives, directories, and/or files that comprise the backup set. This list is created from within the *COREfast* menu or via any text editor, such as the MS-DOS command EDLIN. The catalog scheme adapts well to application-oriented backups.

COREfast can use its proprietary format for high-speed disk backups, either in 360K drives or 1.2Mb AT-style units. The program can alternatively employ preformatted standard MS-DOS media, including other hard disks. It allows data interchange between PC XT and AT computers, if the high-capacity disk formats are

avoided. The program does not make use of multiple disk drives and therefore is not quite as fast as *FASTBACK*.

COREfast version 1.33 corrects problems version 1.30 had with high-density AT-style disk drives under PC-DOS 3.1.

***Intelligent Backup* versus *TakeTwo*.** These two backup programs are discussed together because they are the most supportive. Both offer great flexibility and hand-holding. They'll even recommend what kind of backup you need.

However, *Intelligent Backup* from Sterling Software, is a PC version of a mainframe product. It's big (256K) and it's slower than MS-DOS when copying large files.

TakeTwo, on the other hand, is smaller, faster, and offers a separate program, *Wakeup*, for unattended backups. It's the clear choice between these two. Watch out, though—*TakeTwo* can have problems handling disk errors.

***Backtrack*.** *Backtrack* is another automated-mode backup program. It also has a background mode, coming into action when your computer is idle. It selects files to copy automatically. You don't have to do anything but change the backup medium when needed.

***PDISK*.** *PDISK* is an array of disk utilities. It's fast and reliable with outstanding versatility. For example, a single command can quickly move (not copy) an entire directory tree. It's possible, but not likely, to confuse it if your directories are exceptionally complicated. Its retail price is $145.

Laying Out Your Backup Plan

At this point, you know your options. It's time to make a backup plan that works for you. Use the worksheet provided, making copies as needed.

Take a long look at each application. Applications may be major program areas providing word processing, accounting, spreadsheets, and databases. They could be functional areas, such as Production, Accounting, Distribution, Sales. They could be types of data: Letters, Proposals, Documents, Budgets, Forecasts, Plans, Project Management Data, Mailing Lists, Customer Lists, Inventory, and so on.

Regardless of how you split your data, you need to ask certain questions. What must you back up? How often? Who's going to do

it? Given the size of our files and the desired backup frequency, what medium and type of backup make the most sense? Where should we store the backups? How many generations of backup will we keep for each application?

Study your current directory tree carefully. Does it serve your backup needs well? If you mix programs and data in a single directory, you may want to separate them. If sensitive, rapidly changing data shares a common directory with stable data files, consider splitting them. Is each user's data in his/her own directories? Life will be simpler if it is.

Our Backup Plan for: (Application Area) ______

Directories: ______
Filenames: ______
Backup program name: ______
Where to back them up (medium, volume names, drives): ______

______

______

Must disks be preformatted? ______ How many disks are needed? ______
Where are backups stored? ______
Where are backup log records kept? ______
Additional materials available to use in recovery (printouts, and so on.)

______

Additional precautions, warnings, notes or special instructions:

______

______

______

Master backups
When to backup? ______
Catalog or batch file name? ______
Options? ______
How many disks or tapes are usually needed? ______
Who's responsible? ______
How many generations are kept? ______

Incremental backups
When to backup? ______
Catalog or batch file name? ______
Options? ______
How many disks or tapes are usually needed? ______
Who's responsible? ______
How many generations are kept? ______

Other backups: sensitive files

Directories: ______

Filenames: ______

Backup program name: ______

When to backup? ______

Catalog or batch file name? ______

Options? ______

Where to back them up (medium, volume names, drives): ______

______

Must disks be preformatted? ______

How many disks or tapes are usually needed? ______

Who's responsible? ______

How many generations are kept? ______

Where are backups stored? ______

Where are backup log records kept? ______

Additional materials to use in recovery: ______

______

Additional instructions: ______

______

Reliability

	Yes	*No*
Can standard MS-DOS program/commands read copies?	______	______
Is Verify mode available?	______	______
Is byte-by-byte comparison available?	______	______
Is error correction available?	______	______

	Poor	*Fair*	*Good*
How is the program's handling of bad disks?	______	______	______
How do you rate the program's error reporting?	______	______	______

Price

How much can you afford? ______

What does this program cost retail? ______ discounted? ______

Ease of use

	Yes	*No*
Does it offer the following?		
Installation program or procedure	______	______
Command-line mode	______	______
Menu mode	______	______
Backup catalogs (you define each backup procedure once, including filenames and options, then call it up when needed)	______	______
Disk preformatting	______	______
Online help	______	______
Telephone support	______	______
Regular updates	______	______
Any guarantee	______	______
Preprinted disk labels	______	______

Can it format disks on the fly?	____	____
Can it handle errors without stopping part way through?	____	____
Does it beep when it's time to change disks?	____	____
Does it create a backup log as part of the backup process?	____	____
Does the manual have these qualities?		
Thorough	____	____
Readable	____	____
Indexed	____	____
Does it suggest backup plans or systems?	____	____

Speed

	Fast	*Average*	*Slow*
Formatting speed	____	____	____
Copying speed	____	____	____
Moving speed (if available)	____	____	____

Unit Capacity (per floppy, tape, and so on) ____________ Mb

Flexibility

	Yes	*No*
What backup options are available?		
Image backup	____	____
Complete file-by-file backup	____	____
Backup files after a date	____	____
Backup files before a date	____	____
Backup changed files (archive bit on	____	____
Only replace files with an earlier date	____	____

Other options: ____

____

Can the program change modes		
Background mode	____	____
Unattended (time-set) mode	____	____
Does the program include any other disk-management utilities?	____	____

List those of interest: ____

____

What media can the program work with? ____

Chapter 9

Copy Protection

Copy Protection

Time is money, except when spent copying someone else's programs. Good software takes a lot of time to design, write, debug, and document. Even bad software is expensive to create, package, and market.

Disks and tapes allow you to copy your files or programs quickly and easily. They also provide a simple means of loading files and programs onto your hard disk—or someone else's. Knowing this, software publishers have tried using copy protection as a means to guarantee payments from everyone using their products.

International software piracy is rampant. Application software is sold for $10 per disk in Taiwan and Hong Kong. Many software "rental" firms virtually invite disk copying. Manufacturers have long sought protection of their proprietary rights, using whatever means possible.

When Franklin Computer introduced its clone of the Apple II, Apple Computers stopped them with an injunction enforced by the United States Customs officials. The reason for the action was Apple's allegation that the Franklin's operating system infringed on a copyrighted piece of software.

Phoenix Computer Products, manufacturer of a Basic Input/Output System (BIOS) widely used in low-cost PC clones, went to Taiwan mid-1987 to initiate criminal proceedings against pirate BIOS manufacturers. A similar raid in Canada by a consortium of software publishers late in 1986 netted illegal copies of such high-ticket software as *1-2-3*, *AutoCAD*, and *dBase III*.

So be aware that copying software that's protected may be widespread, but it is also a serious crime.

Copy-protected programs are merely inconvenient to use with a floppy disk–based computer. On hard disk systems, the protection schemes range from inefficient to aggravating to unwieldy. As the remainder of this chapter illustrates, copy-protected software

limits the options of the paying customer. Consumer pressure has forced most software publishers to remove copy protection from much of the software sold within the United States.

> Keep copy-protected software off of your hard disk.
>
> - Given a choice of equivalent products, pick the unprotected brand.
> - If you can get either a protected or an unprotected version of a package, spend the extra money wisely and save yourself some grief by buying the unprotected version.
> - If you must register a copy-protected package prior to receiving an unprotected copy, run from floppy disks until you can install the unprotected copy onto your hard disk.
> - If you must use a copy-protected program, install it onto your hard disk only after you first *unlock* it with a protection-removal program.

Copy-Protection Methods

Software publishers have tried various ways to prevent unauthorized copying or use of their products. Copy protection uses one of three basic methods:

- marked key disk
- programmed hardware key
- hard disk drive identifier

Key disk method. When the protected program runs, it checks to see whether the uniquely-marked original disk is in the drive. The mark is either a predictable error, or predefined disk damage. Errors are typically a sector of nonstandard length, an incorrect cyclic redundancy check (CRC) value, or a mispositioned address mark. Damage takes the form of a permanent hole through the media.

By introducing an apparent error on the disk, the publisher prevents you from using either the COPY command or the DISKCOPY program. Either you *Retry* forever, *Abort* the copy, or *Ignore* the error and make the copy. When you select the *Ignore* option, the copy program attempts to correct the error. The protected software then does not find the expected error and refuses to run.

Hardware key method. A hardware key is a device that is

either installed within the computer or connected to one of the external ports. The key communicates with the protected software.

As the protected software runs, it passes information to the hardware key and receives information back. This information may either be constant—such as a serial number—or variable. One method of variability is using the date or time to scramble the serial number.

During the 1960s the hand-assembled IBM-360 series of mainframe computers contained ROM (Read Only Memory) that supplied the computer's serial number. Mass-produced IBM PCs of the 1980s did not, nor did their clones. Some hardware keys for MS-DOS computers reside on a board inserted into the computer. Other keys serve as an extension cord for the connector to the keyboard, the printer, or modem.

The Micro Channel bus of IBM's new PS/2 line only activates those slots containing boards with a registered product identification code, recorded within an *adapter description file* (ADF). Ostensibly, this optimizes the way the system works with the particular board. This approach is also consistent with IBM's stated "clone-killing" policy: It keeps IBM in control of which third party manufacturers introduce which accessory boards.

Consequently, some manufacturers simply used the same ADF ID numbers as the equivalent IBM products.

Drive identifier method. In the drive-identifier scheme, an installation program writes unique information about the specific hard disk into a hidden file or into the program being installed. The protected program then runs only when it finds the same information.

What is unique about a mass-produced hard disk? Few hard disks are perfect. Every drive contains a list of those areas that are unreliable *(bad tracks).* As Figure 9-1 shows, programs can read the bad track list. The odds are low that two hard disks would have the same list of bad tracks. Various disk drive models respond at different speeds. A program can use both the bad track table and the drive performance timing to identify a specific computer and drive combination.

Figure 9-1. HTEST program reads and displays bad track table.

```
       Kolod Research, Inc.     - HTEST -      IBM PC Hard disk TEST

   Current Cylinder:  613 Current Head:    3  Error page #:    1  of    1

  Cylinder   Head   R/C      Cylinder   Head   R/C      Cylinder   Head   R/C
  --------   ----   ---      --------   ----   ---      --------   ----   ---
     336        0   0Ah         379        0   0Ah         419        0   0Ah
     443        3   0Ah         612        0   0Ah




ENTER . . . (A)-Exit HTEST
            (N)-Next error page (P)-Previous error page  (H)-Error HELP text
```

A variation on this theme is to record the location of the protected software into one or more *invisible files*, or *dams*. *ProLok* uses this technique.

> To determine whether the software is protected:
>
> - Look on the package or manual for the statement *Not copy-protected*. This should be readily visible, as it is a definite selling-point.
> - Check manual's set-up instructions for floppy disk users. If it tells you to store the original disks after making *working disks* via the MS-DOS COPY or DISKCOPY commands, the software is not copy-protected.
> - Check the manual's set-up instructions for hard disk users. The book will tell you to enter a command to start the installation process. Use the DIR (Directory) command to see if that installation command is a batch file. For instance, if the book says *type A:INSTALL C:* look for the file INSTALL.BAT on the disk. If you find it, TYPE or PRINT the batch file and see if it uses only the CLS, ECHO, CD, CHDIR, MD, MKDIR, and COPY commands. If so, the software is not protected.
> - Ask the software publisher or your dealer. Many dealers do not use the programs they sell; beware of relying solely upon the dealer's opinion.

Copy Protection Drawbacks

Nothing is free. Each copy protection scheme has its own costs to the users. Users are inconvenienced, forced to live with inefficiency, or limited in expansion options.

Key disk drawbacks. This method requires that the original disk be present when the protected software runs. On a floppy-based system, the disk was already there. The only problem was that users could not combine two protected programs onto a single disk.

You may remember being amazed at the speed of your new floppy disk–based computer. If so, you probably recall how soon that blazing speed became intolerably slow. You bought your hard disk for speed and convenience. But having to find a key disk and insert it into a drive is neither speedy nor convenient.

MS-DOS is unable to read the key portion of key disks without errors. Some authors of key disk software bypass the operating system and manipulate the disk drive controller chip directly. The timing of such operations is critical. Whenever a computer manufacturer offers a faster computer speed or a different version of the disk drive controller chip, copy-protection programs must be rewritten. There is always a delay after new hardware appears before the protected software can run with it.

Very soon after the introduction of the IBM PC AT, power users discovered that they could easily boost the 6MHz system speed to 8MHz, and sometimes beyond. They merely replaced an inexpensive 12MHz oscillating crystal with one of a higher frequency. Unfortunately, they also discovered that the installation program for Ashton-Tate's *dBase* no longer accepted legitimate disks. If *dBase* had been installed at 6MHz, it would subsequently refuse to run from the hard disk at 8MHz. Once clone manufacturers—and, eventually, IBM—made 8MHz the standard speed, Aston-Tate modified its copy-protection for the higher clock rate. With the advent of 12MHz machines, Ashton-Tate dropped copy protection.

Hardware key drawbacks. When a user moves from floppy disks to a hard disk, he usually finds he can use the computer for more than one task. Since each software package has its unique key, a user of many programs could end up with keys stacked like a Tower of Babel. Also, workers who move from computer to computer during the day (an increasingly common practice) must constantly disconnect and reconnect appropriate keys.

Mass production results in a lowered cost per unit. The first production copies of microprocessor chips often cost hundreds of dollars. Once the nonrepetitive engineering costs of designing and developing the chips have been recovered, costs drop to a few dollars. This is the economy of scale.

Hardware keys can also be inconvenient for manufacturers. Disks are duplicated on special-purpose machines. Operators load stacks of blank disks into hoppers. The copier reads the entire contents of a master disk into computer memory. Then the machine writes multiple copies of that master. This is called *read one, write many*. It's completely automated—until you add serial numbers.

The better disk-duplicating machines can automatically increment serial numbers and insert them into the memory image of the master disk prior to writing it. The duplicators can also print matching numbered disk labels. But uniquely identifying the master image means making only one copy at a time: *Alter one; write one.*

Novell requires that each server running its *Netware* and *Advanced Netware* contain a hardware key. Within this key is the network's serial number. The software allowing a workstation to connect to the network resides on an identically-numbered master boot disk. This start-up software must be copied from the master disk to the local storage of each workstation. If the serial numbers do not match, a workstation is excluded from the network.

Early in 1987, some customers wishing to expand beyond the limits of *Netware* to *Advanced Netware* had to wait several months. Why? Novell was behind schedule in serializing the master disks and hardware keys.

Drive identifier drawbacks. Problems arise with drive identifier software protection when either the computer speed or the disk drive change. What good is a backup if software restored to a replacement disk drive refuses to run? When first introduced, neither the high-density 1.2MB disks of the PC AT nor the 3½-inch disks of the PS/2 could be used with most existing copy-protected software.

During the third quarter of 1986, software vendors using the *Superlock* copy-protection system from Softguard Systems were dismayed to learn that their disks were not running on the new PS/2 hardware. Most of Softguard's clientele used standard PCs to duplicate protected disks. The timing tolerances of the copy-protection code, when run on PS/2 hardware, only accepted disks manufactured

on expensive and exacting commercial disk duplication machines.

Gathering fragmented files into contiguous disk areas *(packing)* can dramatically speed up the operation of a hard disk. This is because the disk drive heads don't have to search all over the drive to load the files (see Chapter 6). Software that uses its disk file address as part of a hidden security file cannot be relocated: It will not run after packing.

Version 1.3 of the *DesqView* control program and the QEMM memory manager enabled computers with the Intel 80386 computer chip to run multiple copies of MS-DOS at once. Computer board manufactures began offering add-in 80386 processor boards to upgrade existing PC XT and PC AT systems. Users had visions of huge *Lotus 1-2-3* spreadsheets running alongside large *dBase III* database programs. Alas, the *Lotus 1-2-3* copy protection refused to run with Intel's 80386 Inboard speedup board installed.

Overcoming Copy Protection

Nature abhors a vacuum. So does the marketplace. As users discovered the drawbacks of copy-protected software, *copy busting* was born. Each copy protection method has a strategy to overcome it.

Overcoming key disks. Since the key disk always has the same error in a standard location, programs that rely on MS-DOS to detect the error can be fooled. A special *Terminate-and-Stay-Resident* (TSR) program can be loaded to monitor all requests by programs for MS-DOS disk activity. Just as keyboard enhancement products translate simple keystrokes into complex keyboard sequences, these TSR *demons* respond to queries about errors at the standard location by pretending that DOS found the expected error. Central Point Systems offers *NOKEY* as part of its *Copy II PC* package. Quaid Software offers the same function with its *RAMKEY*.

Overcoming hardware keys. Hardware keys are too expensive to be absorbed into the cost of any but the most expensive software packages. The keys are not mass produced. They must be individually programmed with a serial number. This protection scheme is being overcome by its lack of acceptance by software publishers or users.

Market-leader Autodesk tried hardware keys during 1986 on a single version of their *AutoCAD* drafting package. User resistance forced them to abandon the keys within a few months.

Overcoming drive identifiers. TranSec offers its *UNlock* programs to disable the copy-protection code within various software packages. As shown in Figure 9-2, TranSec's tools are for specific program versions, as well as generic *ProLok* and *SuperLok* protection. Avoid buying protected software versions that are not listed as a target in at least one of the *UNlock* programs.

Figure 9-2. UNlock removes copy protection from specific software versions.

```
                UNlock Plus 5.0.  [XXXXXXXXXXXX]

                 Specify the program you wish to UNlock:

                  Package                              Versions supported
dBase III                                              1.0, 1.1, 1.2, PLUS or RUNTIME
Framework                                              1.0, 1.1, and II
FASTBACK                                               5.3
Chart Master                                           6.1
Sign Master                                            5.1
Harvard Total Project Manager                          1.10
Think Tank                                             2.0, 2.1
Lotus 1-2-3                                            1.A, 1.A*, 2.0 and 2.01
Dollars & Sense                                        2.0
Symphony                                               1.1
Clipper
Lotus 1-2-3 Report Writer                              1.0
DoubleDos
Managing Your Money                                    1.5, 1.51, 2.0
Microsoft WORD                                         1.15, 2.0, 2.01


 Use Arrow keys to select a package, then press Return to UNlock it.
                        Or press Esc to Exit.
```

Summary

Copy protection never works for the good of the consumer. Sooner or later, the protection scheme will cost you time, convenience, or flexibility. In the authors' experience, businesses are very careful to pay for all their software. Copy-protected software is usually unnecessary. It is a bad medicine that is steadily being pulled from the shelves.

Chapter 10

Recovery

Recovery

Sooner or later, you'll have problems with your hard disk. Hardware degradation, system crashes, or accidental deletions will damage or destroy precious data. Because this experience is so universal, a battery of recovery tools is ready and waiting. This chapter tells what to use, why, when, and how. It covers MS-DOS, public domain, and commercial programs for the entire gamut of catastrophes.

Don't panic. You can recover:

- Lost space
- Erased files
- Bad sectors and damaged files
- Broken or deleted directories
- Accidentally reformatted disk
- Total hardware failure

Lost Space

MS-DOS keeps track of disk space in units called clusters. Cluster size varies from 512 to 8,192 bytes depending on the MS-DOS version and disk type. The file allocation table (FAT) keeps track of what is in which cluster. (Chapters 1 and 3 described all this in more detail.) But what happens if MS-DOS "hiccups" while writing a cluster? Or what if directory pointers are damaged by bad sectors, write errors, system crashes, and so on? One result is clusters which MS-DOS thinks are in use but aren't—*lost* clusters.

The MS-DOS check disk command, CHKDSK, checks directories and recovers lost clusters. The format is

CHKDSK [filename] [/F][/V] [logfilename]

The filename can include a disk, path, and filename, with or without wildcards. Without a filename, CHKDSK checks all directory entries on the current disk. The */V* option lists directories and

files being checked, plus any error messages, as the check progresses. You can redirect this information to a file or printer with this option.

CHKDSK verifies that every FAT entry belongs to a file. If some do not belong, the /F option names and records the stray clusters. If you have chosen /F, the program will ask *nnnn lost clusters found in nnn chains. Convert lost chains to files (Y/N)?*, after completing its search.

When you press Y, CHKDSK creates one file for each chain of lost clusters, placing them in the root directory. It names them FILE0000.CHK, FILE0001.CHK, FILE0002.CHK, and so on. You can then recover the space by erasing these files:

ERA FILE*.CHK

Figure 10-1. CHKDSK's /V option verifies the integrity of the file allocation table.

```
      C:\WS4\WFSM.SYN
      C:\WS4\WFBG.SYN
      C:\WS4\PRINT.TST
      C:\WS4\README.TXT
      C:\WS4\WSINDEX.XCL
      C:\AUTOEXEC.BAT
      C:\AUTOEXEC.CIE
      C:\CONFIG.EMS
      C:\AUTOEXEC.INP
      C:\BACKUP.M_U
      C:\CONFIG.SYS

 21282816 bytes total disk space
    45056 bytes in 2 hidden files
   104448 bytes in 45 directories
 17606656 bytes in 1294 user files
    51200 bytes in bad sectors
  3475456 bytes available on disk

   655360 bytes total memory
   359888 bytes free

All specified file(s) are contiguous.
```

The /F option also attempts to fix any damaged files. However, it's not very good at it. Try running CHKDSK twice. The first time, instead of the /F option, use /V*.*

CHKDSK *.* /V LOG.TXT

You'll find out if any files have major problems. If so, copy them to another disk for further recovery efforts. Then run CHKDSK again with the /F option.

If you give a filename, CHKDSK also tells you how fragmented your files are. When fragmentation is bad, the cleanup programs described in Chapter 5 can reunify files and speed up your system.

A third CHKDSK function is finding and marking bad sectors. You should probably use CHKDSK regularly—once a month or so—for housekeeping. Regular housekeeping could prevent grief later.

Erased Files

It's the end of a long day. You're going to erase one or two old files, then go home. Your finger slips, and you type the wrong filename. Or, worse, your mind slips, and you type:

DEL *.*

What now? As lonely as you might feel at that moment, you are not the first person to make this error. Fortunately, the demand for a remedy for this situation has resulted in a number of disk utility programs which include undelete or unerase features.

"Unerasing" is easy, due to the way MS-DOS erases a file. An erased file stays on the disk. Its directory entry changes—the filename's first letter changes to a ? with the high-order bit set—and the file allocation table is changed to mark the clusters where the file resides as available, but the file itself is not physically erased. Therefore, restoring the file consists of:

- Finding the directory entry
- Restoring its first letter
- Updating the file allocation table

Figure 10-2. Using *Norton*'s QU to recover erased files before their disk space is given to another file.

```
C:\MASM>qu *.asm
QU-Quick UnErase, Advanced Edition, (C) Copr 1987, Peter Norton

Directory of C:\MASM
   Erased file specification: *.ASM
   Number of erased files: 4
   Number that can be Quick-UnErased: 4

   Erased files lose the first character of their names.
   After selecting each file to Quick-UnErase, you will be
   asked to supply the missing character.

   ?bm.asm         4,245 bytes   7:24 pm  Sun Aug 28 83

   ?ompaq.asm      4,248 bytes   7:23 pm  Sun Aug 28 83

   ?path.asm       9,338 bytes  10:06 am  Sat Dec  8 84
'dpath.asm' Quick-UnErased.

   ?os-edit.asm   62,209 bytes   6:06 pm  Thu May  3 84
Enter the first character of the filename:
```

Norton Utilities, Mace Utilities, and *PC Tools* are just a few programs with unerase options. Whatever program you use, it's important to *unerase immediately,* before writing anything on your disk. MS-DOS has marked the erased file's clusters as available for use. Anything written to disk is likely to overwrite your file. As Figure 10-2 depicts, timely recovery is easy.

Bad Sectors

Sectors are a disk's physical building blocks. Bad sectors can be caused by physical damage to the disk from dirt, dust, a head crash, wear, and so on. Bad sectors can also result from *logical,* or *nonphysical recordkeeping* damage. Early discovery is the best protection. MS-DOS can mark identified bad sectors so no files attempt to use them.

The CHKDSK command identifies and marks bad sectors. CHKDSK was discussed earlier in this chapter. For some reason, CHKDSK doesn't catch all bad sectors. Many commercial utilities, including *Mace Utilities, Norton Utilities,* and *PC Tools,* do a better job. All these programs will find as well as mark bad sectors. Figure 10-3 shows *Norton*'s DT program searching the entire disk for

Figure 10-3. Using *Norton's* DT to Mark Bad Sectors and Find Corrupted Files

```
C:\NORTON>dt
DT-Disk Test, Version 4.00, (C) Copr 1984-87, Peter Norton

Select DISK test, FILE test, or BOTH
Press D, F, or B ... D

During the scan of the disk, you may press
BREAK (Control-C) to interrupt Disk Test

Test reading the entire disk C:, system area and data area
  The system area consists of boot, FAT, and directory
    No errors reading system area

  The data area consists of clusters numbered 2 - 10,393
    5,668th cluster read error: already marked as bad; no danger
    5,669th cluster read error: already marked as bad; no danger
    5,670th cluster read error: already marked as bad; no danger
    5,671st cluster read error: already marked as bad; no danger
    5,672nd cluster read error: already marked as bad; no danger
    5,916
```

corrupted sectors. *Mace*'s HTEST utility is even better at finding bad sectors, but it doesn't mark them. Check regularly for bad sectors and lost space.

Fixing Damaged or Corrupted Files

A *damaged* file sits on a bad sector or sectors. MS-DOS can't read or write it properly. A *corrupted* file contains incorrect information. MS-DOS can read and write to it, but its content is wrong.

To restore a damaged file you must:

- Move it to another physical location
- Fix any content errors

To restore a corrupted file, only perform the second step above.

Most programs that find bad sectors can also move files sitting on them. This is true of CHKDSK, *Mace, Norton Utilities,* and *PC Tools*. They move the file to another location and mark the sectors as bad. In moving, they attempt to read the file's contents, but some information may be lost or distorted.

MS-DOS's RECOVER command is another way to move damaged files and mark bad sectors. RECOVER simply skips anything it can't read. *Be sure to give a filename when using RECOVER.* If not,

RECOVER tries to fix the entire disk, often with disastrous results.

When your damaged file is moved, it is a corrupted file. In other words, you now have a file whose contents may be wrong. It's up to you to fix it. The method for fixing the corrupted file depends on the type of file.

Don't fix program or support files. Program files include not only .EXE or .COM files, but also any overlays they call. Most program files are compiled, unreadable code. It's almost impossible for you to make sense of them. Furthermore, even the slightest error in a program file can be catastrophic. For example, the program might mistakenly write data to the wrong disk location.

For both reasons, don't even try to fix a program file unless it's ASCII text or source code. Even then, it isn't recommended. You probably have a copy of the original program file on a floppy disk somewhere. (If not, make copies now). To fix a corrupted program or support file, simply copy the files off the floppy disk onto the hard disk. Likewise, restore other support files from backup if possible.

Fixing text files. Copy the file before trying to fix it. Use your word processor or text editor to examine the file. If the file contains source code or anything other than a letter or a document, use nondocument mode. You may find damage to one or more 512-character chunks and extra garbage at the end. Replace the damaged chunks from a backup if you have one. Remove the extra garbage. Figure 10-4 illustrates using the DEBUG program to replace a bad sector within a text file with ASCII spaces.

What if you're missing part of your file? Chances are you have a stray end-of-file (EOF) character in your text. Most editors ignore anything after the EOF character. Use the DOS DEBUG program or *Norton Utilities'* NU to find and change it. The end-of-file marker is Ctrl-Z (ASCII code 26). Change it to another character, say, a space (ASCII code 32). If you're not comfortable using DEBUG or NU, find an experienced friend to help.

Fixing a nontext data file. Make a backup copy of the corrupted file before you start to fix it. Replace it with a backup copy if possible. If it's an encrypted file, you may want to start from a nonencrypted backup. If you don't have backups, use DEBUG or another nontext editor such as *Norton Utilities'* NU program.

Figure 10-4. DEBUG can replace corrupted text with spaces to allow further repair with a text editor.

```
C:\CIE\TEMP>debug main.h
-d 250 31f
4BF6:0250  76 69 72 6F 6E 6D 65 6E-74 73 0D 0A 2A 20 20 20   vironments..*
4BF6:0260  20 20 20 20 50 4F 20 42-6F 78 20 34 35 37 2C 20       PO Box 457,
4BF6:0270  4C 61 20 4D 65 73 61 2C-20 43 41 20 39 32 30 34   La Mesa, CA 9204
4BF6:0280  E5 E5 E5 E5 E5 E5 E5 E5-E5 E5 E5 E5 E5 E5 E5 E5   ................
4BF6:0290  E5 E5 E5 E5 E5 E5 E5 E5-E5 E5 E5 E5 E5 E5 E5 E5   ................
4BF6:02A0  E5 E5 E5 E5 E5 E5 E5 E5-E5 E5 E5 E5 E5 E5 E5 E5   ................
4BF6:02B0  E5 E5 E5 E5 E5 E5 E5 E5-E5 E5 E5 E5 E5 E5 E5 E5   ................
4BF6:02C0  E5 E5 E5 E5 E5 E5 E5 E5-E5 E5 E5 E5 E5 E5 E5 E5   ................
4BF6:02D0  E5 E5 E5 E5 E5 E5 E5 E5-E5 E5 E5 E5 E5 E5 E5 E5   ................
4BF6:02E0  E5 E5 E5 E5 E5 E5 E5 E5-E5 E5 E5 E5 E5 E5 E5 E5   ................
4BF6:02F0  E5 E5 E5 E5 E5 E5 E5 E5-E5 E5 E5 E5 E5 E5 E5 E5   ................
4BF6:0300  2D 2D 2D 2D 2D 2D 2D 2D-2D 0D 0A 2A 20 57 48 49   ---------..* WHI
4BF6:0310  43 48 20 7C 20 20 20 57-48 45 4E 20 20 20 7C 20   CH ¦   WHEN   ¦
-f 280 2ff, 20
-w
Writing 0B00 bytes
-q

C:\CIE\TEMP>
```

Lost or Damaged Directories

What if a directory sits on a bad sector? Or what if the system crashes while writing a directory? In a sense, fixing a damaged directory is no different from fixing any damaged file. However, the stakes are much higher. Without a directory pointer, MS-DOS doesn't think you have a file. If you damage a directory, you can lose many files instantly.

The general file recovery programs discussed in the sections on Bad Sectors and Lost Files can sometimes find and fix damaged directories. Two types of specialized programs may be even more useful.

First are programs geared specifically to finding and recovering directories. For example, *Norton Utilities'* UD (Unremove Directory) option searches for recoverable directories, then walks you through possible file listings, as in Figure 10-5. Once you have recovered the directory, you can unerase its files with QU. The UD program is most useful when

- You know exactly what files were in the directory
- The directory was in a sorted sequence

Figure 10-5. Using *Norton*'s UD to recover a deleted directory so its files may be unerased.

```
C:\CIE>ud
UD-UnRemove Directory, Version 4.00, (C) Copr 1984-87, Peter Norton

Directory of C:\CIE
   Removed directory specification: *.*
   Number of removed directories: 1
   Number that can be UnRemoved: 1

   ?EMP             <DIR>       4:35 am  Mon Jul 20 87
Enter the first character of the filename: t

   ?asyc.h         1,152 bytes  12:00 am
   ?cntl.h           768 bytes  12:00 am
   ?ain.h          2,816 bytes  12:00 am
   ?ortab.h        2,176 bytes  12:00 am
   ?indow.h        4,096 bytes  12:00 am
   ?os.h             768 bytes  12:00 am

Files included in C:\CIE\TEMP

'TEMP' UnRemoved

C:\CIE>
```

If neither is true, UD may not know when it's reached the last directory entry. However, you will probably know when you start seeing useless file names. In the worst case, you may have to delete any extra files you included.

A second type of utility can be especially useful for finding and fixing directories. These utilities keep their own backup copies of your directories. For example, *Mace Utilities* uses its RXBAK utility to keep its directory tables in a file named BACKUP.M_U. Mace can use this information to rebuild one or more of your directories.

Recovering a Reformatted Disk

What if your new assistant unknowingly reformats your entire disk? As discussed in Chapter 7, this is surprisingly easy to do. All your directories and files will disappear. Luckily, on most machines and with most versions of MS-DOS, the directories and files aren't really removed from the disk. Remember that DEL and ERASE simply change directory entries without really wiping off the files. Likewise, FORMAT simply clears the file allocation table and the

root directory. Everything else stays where it was, waiting to be found.

As with lost directories, two kinds of help await you. The first type are utilities which keep their own directory/file tables, separate from those used by MS-DOS. The second type are programs which can unformat "on the fly" by knowing what to look for.

DSRecover and *Mace Utilities* maintain their own directory backups. If your disk is reformatted, they can find their own tables, then use them to recover the rest of the disk. In such cases, disks can be completely recovered up to the time of the last backup. Figure 10-6 displays the *Mace* recovery menu.

Figure 10-6. Mace allows recovery from drastic errors.

```
F1--HELP                      F2--DIR
F3--Restore BOOT              F4--CHKDSK
F5--Restore BACKUP.M_U        F6--RECLAIM

F9--UnDELETE                  F10-UnFORMAT

         Select one!  [Esc to escape]
```

Mace also offers an on the fly option. If you buy the program after you have reformatted and don't have the backup directory available, you can still recover most of your disk. The program uses every trick it can muster to recognize both files and directories. For example, the first entries in any subdirectory are the . and .. pointers to the current and parent directories, respectively. *Mace* searches for properly spaced . and .. text as signs of possible directories.

A few versions of MS-DOS are less forgiving. Compaq MS-DOS 3.1 and AT&T MS-DOS 2.11 are among those that wipe the entire disk when they format. In such cases, the only recovery is a recent, complete backup of your entire disk.

Signs of Impending Hardware Failure

It's possible for a drive to work perfectly one day, then to be useless the next. However, it's not likely. Usually you can spot trouble coming if you're willing to look and listen.

If any symptoms appear, backup more frequently. Be sure you always have at least two full backups, made on different days.

Keep both old and new backups. If random damage is affecting files, you don't know how far back you must go to find a good version of any file.

Also, use a program such as *Mace Utilities'* RXBAK which duplicates the boot sector, file allocation table, and root directory. Copy this backup information (called BACKUP.M_U) offline daily.

Here are common warning signals, from least to most serious. Even the least serious is important and dangerous. Don't ignore any warning.

- Grinding noises while reading or writing the disk. This noise usually comes from repeated attempts to read or write marginal sectors. Probably this noise will accompany other symptoms of bad sectors as discussed below.
- An increase in bad sectors. A small number of bad sectors on a new disk is normal and no cause for alarm. Likewise, a small number of new bad sectors is acceptable. You can expect roughly one error per ten million sectors written. Any significant number of new bad sectors suggests physical shock, misalignment, wear, or intermittent electronic problems. In other words, they indicate impending trouble. Watch out.
- An annoying high-pitched whine that wasn't there when the drive was new. This is a common sign of spindle or bearing wear. It results from friction in the drive. It precedes total failure of the spindle or bearings.
- MS-DOS cannot read or write a sector. Usually this message is a sign of malfunction in the drive's electronic circuitry. This is extremely serious. Electronic problems can send good data to bad locations. Directories, the file allocation table, or part of cylinder zero can be overwritten, with disastrous consequences.
- Disk boot failure. This MS-DOS error message results from damage to the operating system. Restoring MS-DOS is quite easy. However, the real question is: What caused the damage? Whatever struck once may strike again.
- Bad partition table. This message shows damage to the master boot record on cylinder zero. To recover you must reformat cylinder zero. This can be done. If you have been using *Mace* or a similar utility that backs up system information, you can probably also recover your data to the point of last backup. However, you must understand that losing cylinder zero, even temporarily, is a first class emergency.

- *Drive Not Ready* prompt. MS-DOS does not recognize your drive. Either you have a hardware failure or your partition information is lost. It doesn't get any worse than this.

> To restore the boot record, recopy MS-DOS onto your hard disk.
>
> - Start the system with a bootable (system) floppy disk in drive A. It should contain your current version of MS-DOS.
> - Copy the operating system onto drive C with the SYS command
>
> **SYS C:**
>
> - Copy COMMAND.COM onto drive C with the COPY command
>
> **COPY COMMAND.COM C: \ /V**

> Here's how to restore cylinder zero.
>
> - Turn off your machine. Wait a few seconds, then turn it back on. This may be enough to clear the error. If so, go to the last step.
> - Run *Mace Utilities* from a floppy disk. Select the Restore Boot Sector option from the utility menu. If *Mace* runs without errors, try booting from your hard disk. If you succeed, go to the last step.
> - Run the DOS FDISK command from a floppy. This recreates the master boot record. If FDISK is error-free, skip the next step.
> - If FDISK doesn't work, run a physical formatting program on cylinder zero (only cylinder zero). You can use the IBM *Advanced Diagnostics* or *Mace*'s HFORMAT program. Then repeat the third step.
> - Run *Mace* from a floppy to restore BACKUP.M_U, the backup system information. Select Restore BACKUP.M_U from the utilities menu.
> - Make a complete backup of your entire system. If this problem occurs again, you want to be ready.

Here are suggestion on Recovering from *Drive Not Ready* or *Invalid Disk Specification.*

- Run a diagnostic program such as the diagnostics on your MS-DOS *Guide to Operations* disk, IBM's *Advanced Diagnostics,* or *Mace*'s HTEST and HFORMAT.
- If the drive responds, back it up at once using a new set of backup media. Then continue the diagnostics.
- If the drive doesn't respond, be grateful that you have been following your backup plan (see Chapter 8). Take the drive and controller card in for servicing.

Servicing a Drive

In time your disk will die. You'll have to remove it from your machine or bring your entire system to the repair shop. After servicing, you will probably have to restore all your directories and files. Here's hoping you have good backup when it happens.

When servicing a hard disk drive:

- The fatal problem may be in the disk controller card. If you have only one hard disk drive, you do not know which component failed. Send both the drive and the controller for service.
- If possible, park the drive before removing it or moving your system. Some drives automatically move the read/write head to a landing zone when they power off. Others do not, but rely instead upon a program to do this. Chapter 3 gives details.
- Consider having a spare drive. It may be cheaper to have a spare than to suffer long periods of down time while you are without your computer. If you find that your spare drive also fails, you can probably trace the problem to a bad controller card.
- If your system is under warranty, let the repair person know. Sometimes warranty repairs go to the head of the line. Some repairs take less than an hour, but you may have to wait two weeks for your turn. Also, ask if you can get a replacement drive instead of having to wait for yours to come back.
- If your hard disk drive isn't under warranty or you have a choice of warranty repair sites, check your choices carefully. Ask friends and check with the Better Business Bureau. Most shops are reliable, but you may hear some horror stories.
- If you must ship your hard disk drive, pack it carefully in the original packing materials if you have them. Insure it.

In Conclusion

Problems, breakdowns, and repairs: These are unpleasant topics. And they don't come up often because problems don't happen often. But they do happen eventually and must be dealt with. The positive message from this chapter is: You are not a passive, helpless victim. Almost always, you can spot trouble while it's still small. Regular checkups, good backups, and timely concern can put you back in control. Problems don't have to become catastrophes. You're in charge.

To help you stay in charge, use the following Recovery Checklist. Take time now to fill it out and close the gaps in your recovery readiness.

System: ____________________

Checklist completed by: ____________________

Date: ____________________

Warning Signals

Is everyone using this system aware of common trouble signs: high pitched whine, grinding noises, bad sectors, read or write errors, and so on?

How can you help people remember these warning signals?

____

Lost Space and Bad Sectors

How often/when will you run CHKDSK to recover lost space (for instance, the first Friday of the month)? ____________________

Who will do it? ____________________

Do you have a better program than CHKDSK to find/mark bad sectors? Program name: ______

____

How often/when will you run it? ____________________

Who will do it? ____________________

System File Backup

Do we have a program, such as *Mace Utilities* or *DSRecover*, that copies the file allocation table, root directory, and other system information from cylinder zero?

Program name: ____________________

Backup file name: ____________________

Does our startup program update this file? ____________________

Do our other batch files update this file? ____________________

____

When do we copy this file offline? ____________________

Recovering Erased Files

Program name: ______________________________

Online program location: ______________________________

Offline program location: ______________________________

Documentation location: ______________________________

Our resident expert on this function: ______________________________

Recovering Damaged Directories

Program name: ______________________________

Online program location: ______________________________

Offline program location: ______________________________

Documentation location: ______________________________

Our resident expert on this function: ______________________________

Recovering Deleted Directories

Program name: ______________________________

Online program location: ______________________________

Offline program location: ______________________________

Documentation location: ______________________________

Our resident expert on this function: ______________________________

Recovering a Reformatted Disk

Program name: ______________________________

Online program location: ______________________________

Offline program location: ______________________________

Documentation location: ______________________________

Our resident expert on this function: ______________________________

Recovering Cylinder Zero

Program name: ______________________________

Online program location: ______________________________

Offline program location: ______________________________

Documentation location: ______________________________

Our resident expert on this function: ______________________________

Appendix A

Do-It-Yourself Software Toolkit

Do-It-Yourself Software Toolkit

This appendix presents some potent software tools that you can create. The following pages show you how to use text files and the MS-DOS DEBUG.EXE utility to make programs.

The discussion begins with an explanation of the theory, illustrated by an example. Then it continues with a look at a series of small, useful programs. A narrative description is provided for each program, along with an example showing how to use the program. Finally, the program's complete source code is listed.

To turn these listings into disk files, you need a bit of patience, a keen eye, a steady hand, and a nondocument text editor (such as the MS-DOS EDLIN program). Once you have transcribed the text onto your hard disk's \SCRATCH directory, you'll also need the MS-DOS DEBUG utility.

Many of the following programs are essential parts of the batch file menu system described in Appendix C.

Input Scripts and DEBUG

DEBUG.EXE is a special type of editor program. Rather than editing text files, DEBUG edits program and data files. DEBUG contains an Assembler, a language translator that turns specially-prepared text (*source code*) into machine-executable program instructions *(binary code).*

DEBUG can redirect its keyboard input (*stdin*) and its screen output (*stdout*). This means that other programs or data files may supply the keystrokes or store the screen characters. The listings that follow are redirection input files. They contain the keystrokes for DEBUG that you would normally provide from the keyboard when creating the various programs.

DEBUG is an external command. When you want to run it, you need to either include its directory within your standard program search path (as discussed in Chapter 3), place a copy in your current directory, or specify its drive and subdirectory on the command line.

IBM, second-guessing how customers use their products, has decided that only programmers need DEBUG. IBM has excluded all references to DEBUG from the DOS manual for PC-DOS version 3.3. It's described in the *DOS Technical Reference* manual for version 3.3, which costs $85.

Fortunately, the program is still provided with the system software. Other versions of MS-DOS and PC-DOS describe DEBUG in their users' reference manuals.

The input scripts tell DEBUG what to do, so you don't have to know how to run the program yourself. If you're curious, though, the DEBUG commands used are:

- A(ssemble) translates Intel 8086 assembley language instructions into the corresponding binary machine code; it remains in this mode until it finds an empty input line.
- N(ame) indicates which file will be read or written.
- R(egister) queries and sets certain indices using hexadecimal. CX is the count register and indicates the size of a file.
- L(oad) reads the specified file into computer memory (starting at location CS:0100) and sets register CX to the number of bytes in the file.
- W(rite) copies the specified file from computer memory (starting at location CS:0100) to disk. It writes as many bytes as are indicated by register CX.
- Q(uit) ends DEBUG and returns control to MS-DOS.

Figure A-1 illustrates a DEBUG session. Both the program's prompts and the operator's responses are visible. The onscreen activity shows creating BELL.COM, a ten-byte program that sounds an audible tone at the console. It does this by sending the ASCII bell character (07H) to the video controller software in read only memory (ROM).

Figure A-1. A DEBUG Session to Create the BELL.COM Program

```
C:\SCRATCH>debug
-a 100
39AC:0100 mov ax,e07
39AC:0103 mov bx,0
39AC:0106 int 10
39AC:0108 int 20
39AC:010A
-r cx
CX 0000
:a
-n bell.com
-w
Writing 000A bytes
-q

C:\SCRATCH>
```

If you keep track of the keys the operator pressed during the DEBUG session of Figure A-1, you end up with the file listed in Program A-1. BELL.SCR contains the keystrokes necessary to have DEBUG make the BELL.COM program. Notice that a single empty line preceeds the *R CX* line; this ends the A(ssemble) command in the first line.

Program A-1. The script BELL.SCR contains the necessary keystrokes for DEBUG.EXE to create the BELL.COM program.

```
A    0100
MOV  AX,0E07
MOV  BX,0000
INT  10
INT  20

R    CX
000A
N    BELL.COM
W
Q
```

Figure A-2 shows the screen display after redirecting the BELL.SCR into the DEBUG utility. As you can see on the top line of screen text, you make DEBUG use the script by typing **DEBUG < BELL.SCR**.

Figure A-2. An input script supplies all keystrokes in response to the MS-DOS input redirection operator (<).

```
C:\SCRATCH>debug < bell.scr
-A      0100
39AC:0100 MOV    AX,0E07
39AC:0103 MOV    BX,0000
39AC:0106 INT    10
39AC:0108 INT    20
39AC:010A
-R      CX
CX 0000
:000A
-N      BELL.COM
-W
Writing 000A bytes
-Q

C:\SCRATCH>
```

Compare the screens in Figures A-1 and A-2. Notice that they are equivalent. The advantages of using an input script instead of keying directly into DEBUG are:

- DEBUG's text editing is less flexible than general-purpose text editors (with the exception of EDLIN).
- Script files isolate the useful instructions from the chatter of a DEBUG session.
- You don't have to rekey everything to rerun a script.
- Text scripts are easy to communicate and share.

The remainder of this appendix describes software tools that you can create using DEBUG input scripts as listed.

Console Tools

The following software tools provide control over your console display. They are useful whether or not you are running batch files. The console-oriented tools are:

- COLOR.COM, which sets the display colors; defines the text, screen, and border colors; supports displays up to 240 lines deep and 240 columns wide; allows paper mode (black-on-white) monochrome.
- CURSOR.COM, which makes the screen cursor into a block, an underline, or nothing at all; positions the cursor; and reports its screen location for batch menus.

COLOR.COM provides control over your display's colors. It is a stand-alone program, not requiring support of additional software (such as the ANSI.SYS device driver).

The COLOR program varies its behavior according to how it is invoked. It searches the tail of its command line. If it finds one, two, or three decimal numbers separated by commas, it uses them as the text, screen, and border colors respectively. Table A-1 lists the numbers and colors.

Table A-1. COLOR.COM screen colors. Text and border colors range from 0–15. Screen color may be 0–7.

Code	Hue	Contrast
0	Black	White
1	Blue	Brown
2	Green	Magenta
3	Cyan	Red
4	Red	Cyan
5	Magenta	Green
6	Brown	Blue
7	White	Black
8	Dark Gray	White
9	Bright Blue	Brown
10	Bright Green	Magenta
11	Bright Cyan	Red
12	Bright Red	Cyan
13	Bright Magenta	Green
14	Yellow	Blue
15	Bright White	Black

If you don't specify anything on the command line, COLOR.COM sets the screen to its default colors and then displays a usage summary. For a color screen, the default colors are green letters a on a black background. Monochrome displays are set to light-on-black.

If you indicate a single number, it defines your text color. The screen and border then use the appropriate contrast color from the third column of Table A-1. To get yellow letters on a blue background, run the program with the command **COLOR 14**. For monochrome displays, any nonzero color becomes white. Therefore, the preceeding command would cause light-on-black with an IBM Monochrome Display Adaptor (MDA) or a Hercules Graphics Card (HGC). To get a paper mode display of black letters upon a light background, invoke the program as **COLOR 0**.

If you don't like the automatically-selected contrast color (COLOR 2, which selects green-on-magenta, is garish), select both the text and screen colors with two numbers separated by a comma and no embedded spaces. The second number selects the color of the background. To get dark blue letters on a cyan screen, run **COLOR 1,3.**

The IBM Color Graphics Adaptor (CGA) allows separate control over its border color. The Enhanced Graphics Adaptor (EGA) maintains a black border; monochrome display cards (MDA, HGC) set the border color to be the same as the screen color. The border is the portion of the display that lies outside of the text area. If you do not specify a third color number, COLOR.COM makes the border the same color as the screen. If you have a CGA and want red letters on a white background with a blue border, type **COLOR 4,7,1.**

Unlike the intrinsic MS-DOS CLS (clear screen) command, COLOR.COM works with video adaptors that provide more than the standard 25 lines of 80 characters. COLOR.COM uses the built-in ROM BIOS to set the screen colors. Video board manufacturers supply ROM BIOS extensions or device drivers to support the higher-than-normal screen densities they allow you to select.

The ROM BIOS begins numbering screen rows and columns from zero, so a 25-line screen ranges from row zero to row 24, or one less than the total number of rows. Similarly, the columns run from column zero to one less than the screen width.

The COLOR.SCR script tells the BIOS to fill the screen with the colors you have selected, starting at row zero, column zero, and going to row 239, column 239. If this overkill value causes problems for your hardware, use another value from the list in Table A-2.

Table A-2. End-of-Screen ROM BIOS Values for Various Screen Sizes

Screen Size *Rows × Columns*	**BIOS Value**
25 × 80	194F
25 × 132	1983
43 × 80	2A4F
43 × 132	2A83
60 × 80	3B4F
60 × 132	3B83
240 × 240	F0F0

Program A-2 lists the DEBUG script to create COLOR.COM. If you decide to change the size of the screen being cleared, your new value goes into the line near the end of the script. It contains a comment beginning with a semicolon (;). Replace the F0F0 on that line with the appropriate value from Table A-2, and alter the comments accordingly.

Program A-2. COLOR.SCR

```
A       0100
JMP     03DB
DB      0D,0A,0D,0A,0D,0A
DB     'COLOR Screen Setup by Steven Fisher CDP',0D,0A
DB     'Version 1.3 COLOR Screen Setup (C) 1987',0D,0A
DB     'COMPUTE! Publications, Greensboro, NC',0D,0A,0D,0A
DB      'This program sets the text, screen, and',0D,0A
DB      'border colors for the IBM PC. You give',0D,0A
DB      'it 1, 2, or 3 numeric values separated',0D,0A
DB      'by commas. The first value is the text',0D,0A
DB      'color. The second value is the screen',0D,0A
DB      'hue. The third is the border color.',0D,0A,0D,0A
DB      'If only the text color is given, both',0D,0A
DB      'the screen and the border colors will',0D,0A
DB      'be a contrasting hue. If only text',0D,0A
DB      'and screen colors are specified, the',0D,0A
DB      'border and screen are the same color.',0D,0A,0D,0A
DB      'Monochrome video adaptors always use',0D,0A
DB      'the text color to determine the screen',0D,0A
DB      'border colors. Monochrome text is',0D,0A
DB      'either black (0) or white (nonzero).',0D,0A,0D,0A,'$'
DB      00,0A,07,00,00,00,07,00,00,00,00
PUSH    DS
SUB     AX,AX
PUSH    AX
MOV     AH,0F
INT     10
CMP     AL,07
JZ      03F1
AND     AL,03
MOV     [03D2],AL
MOV     BYTE PTR [03D6],02
MOV     SI,0080
MOV     DI,03D9
CLD
MOVSB
```

```
SUB     AH,AH
MOV     AL,[03D9]
MOV     CX,AX
MOV     BX,03DA
CMP     CX,+00
JZ      043C
MOVSB
DEC     DI
MOV     AL,[BX]
CMP     AL,30
JB      0433
CMP     AL,39
JA      0433
MOV     BYTE PTR [03D4],01
SUB     AH,AH
MOV     AL,[03D5]
MUL     BYTE PTR [03D1]
MOV     [03D5],AL
MOV     AL,[BX]
SUB     AL,30
ADD     AL,[03D5]
MOV     [03D5],AL
JMP     043A
NOP
CMP     AL,2C
JNZ     043A
CALL    044A
LOOP    0408
MOV     AL,[03D4]
CMP     AL,00
JZ      0446
CALL    044A
CALL    0489
RETF
MOV     AL,[03D3]
INC     AL
MOV     [03D3],AL
CMP     AL,02
JZ      046D
JA      0472
MOV     AL,[03D5]
AND     AL,0F
MOV     [03D6],AL
NOT     AL
```

```
AND     AL,07
MOV     [03D7],AL
MOV     [03D8],AL
JMP     047E
NOP
MOV     AL,[03D5]
JMP     0462
CMP     AL,03
JNZ     047E
MOV     AL,[03D5]
AND     AL,1F
MOV     [03D8],AL
MOV     BYTE PTR [03D5],00
MOV     BYTE PTR [03D4],00
RET
MOV     AH,0F
MOV     AL,[03D2]
INT     10
MOV     AL,[03D2]
CMP     AL,07
JNZ     04AD
MOV     AL,[03D6]
CMP     AL,00
JZ      04A0
MOV     AL,07
MOV     [03D6],AL
NOT     AL
AND     AL,07
MOV     [03D8],AL
MOV     [03D7],AL
MOV     AL,[03D2]
CMP     AL,07
JZ      04BF
MOV     AX,0B00
MOV     BH,00
MOV     BL,[03D8]
INT     10
MOV     AX,0200
MOV     DX,0000
MOV     BH,00
INT     10
MOV     AX,0600
MOV     BH,[03D7]
MOV     CL,04
```

```
SHL     BH,CL
MOV     BL,[03D6]
OR      BH,BL
MOV     BL,00
MOV     CX,0000
MOV     DX,F0F0      ; 240 lines, 240 cols max
INT     10
MOV     AL,[03D3]
CMP     AL,00
JNZ     04F3
MOV     AX,0900
MOV     DX,0107
INT     21
RET
R       CX
03F4
N       COLOR.COM
W
Q
```

CURSOR.COM: Screen location. CURSOR.COM provides control over your screen's blinking underline cursor. It is a stand-alone program, not requiring support of additional software (such as the ANSI.SYS device driver).

The CURSOR program varies its behavior according to how it is invoked. It searches the tail of its command line. If it finds a recognized keyword, it responds accordingly. The CURSOR keywords and their effects are:

- FULL turns the cursor into a full-character block.
- BIG turns the cursor into a full-character block.
- SMALL turns the cursor into an underline.
- YES turns the cursor into an underline.
- NO makes the cursor invisible.
- HIDE makes the cursor invisible.
- ROW sets ERRORLEVEL to the cursor's Row.
- COL sets ERRORLEVEL to the cursor's Column.
- *rr cc* locates the cursor at row (*rr*) 0–255 and column (*cc*) 0–255, where row 0, column 0 designates the upper left corner of the screen.

CURSOR only looks for the first character of a keyword, so *CURSOR F* is equivalent to *CURSOR FULL*. Other characters are ignored, but if an illegal character or no character follows the CURSOR command, the program will display its help screen.

The MS-DOS internal command CLS erases the PC screen. It also retains the current cursor appearance for the new screen. If you find the standard cursor (a blinking underline) difficult to spot in a screen full of text, you may want run the batch file in Figure A-3. It turns the cursor into a full-character block until some other software (such as MODE) redefines it.

Figure A-3. CLR.BAT is a small job stream that clears the screen and then changes the cursor to a full-character block.

```
echo off
REM clear screen and use block cursor
cursor.com full
cls
```

CURSOR.COM uses ROM BIOS routines to locate the text cursor anywhere from row 0, column 0; to row 255, column 255. Video board manufacturers supply ROM BIOS extensions or device drivers to support the higher-than-normal screen densities they allow you to select. If you specify a location outside the boundaries of your display device (255 255, for instance), the cursor and ensuing text will be invisible. CURSOR.COM also uses the video controller ROM to control the size of the cursor. Program A-3 contains the CURSOR.SCR script.

Program A-3. CURSOR.SCR

```
A     0100
JMP   034A
DB    'CURSOR Version 1.0 by Steven Fisher CDP as of
      08/24/87'0D,0A
DB    '(C) 1987 COMPUTE! Publications, Greensboro, N
      Carolina'0D,0A
DB    'Usage summary:',0D,0A
DB    ' CURSOR BIG     turns on block cursor'0D,0A
DB    ' CURSOR COL     sets ERRORLEVEL to column',0D,0A
DB    ' CURSOR FULL    turns on block cursor',0D,0ADB
DB    ' CURSOR HIDE    turns off cursor',0D,0A
DB    ' CURSOR NO      turns off cursor',0D,0A
```

```
DB    ' CURSOR ROW      sets ERRORLEVEL to row',0D,0A
DB    ' CURSOR SMALL    turns on underline cursor',0D,0A
DB    ' CURSOR YES      turns on underline cursor',0D,0A
DB    ' CURSOR rr cc    puts cursor at row (rr) and col
                        (cc)',0D,0A
DB    '                 where rr and cc range from
                        0-255',0D,0A,'$'
MOV   AX,CS
MOV   DS,AX
MOV   BX,0080
INC   BX
MOV   AH,[BX]
CMP   AH,0D
JZ    0393
CMP   AH,20
JZ    0351
CMP   AH,30
JB    0393
CMP   AH,39
JBE   03D5
AND   AH,5F
CMP   AH,59
JZ    03BD
CMP   AH,53
JZ    03BD
CMP   AH,48
JZ    03B1
CMP   AH,4E
JZ    03B1
CMP   AH,42
JZ    03B7
CMP   AH,46
JZ    03B7
CMP   AH,52
JZ    039F
CMP   AH,43
JZ    03A8
MOV   DX,0103
MOV   AH,09
INT   21
MOV   AL,01
JMP   042F
MOV   AH,03
INT   10
```

```
MOV   AL,DH
JMP   042F
MOV   AH,03
INT   10
MOV   AL,DL
JMP   042F
NOP
MOV   CX,2000
JMP   03CE
NOP
MOV   CX,010C
JMP   03CE
NOP
MOV   AH,0F
INT   10
CMP   AL,07
JNB   03CB
MOV   CX,0607
JMP   03CE
NOP
MOV   CX,0B0C
MOV   AH,01
INT   10
JMP   042D
NOP
MOV   DX,0000
DEC   BX
INC   BX
MOV   AH,[BX]
CMP   AH,20
JZ    0400
CMP   AH,30
JB    0393
CMP   AH,39
JA    0393
MOV   AL,DH
ADD   AL,AL
MOV   DH,AL
ADD   AL,AL
ADD   AL,AL
ADD   AL,DH
MOV   DH,AL
SUB   AH,30
ADD   DH,AH
```

```
JMP   03D9
INC   BX
MOV   AH,[BX]
CMP   AH,0D
JZ    0427
CMP   AH,30
JB    0393
CMP   AH,39
JA    0393
MOV   AL,DL
ADD   AL,AL
MOV   DL,AL
ADD   AL,AL
ADD   AL,AL
ADD   AL,DL
MOV   DL,AL
SUB   AH,30
ADD   DL,AH
JMP   0400
MOV   BH,00
MOV   AH,02
INT   10
XOR   AL,AL
MOV   AH,4C
INT   21

R     CX
0333
N     CURSOR.COM
W
Q
```

Batch Tools

The following software tools are designed to enhance batch processing. They use the enviroment area or the system ERRORLEVEL. The batch-oriented tools are:

- GETREPLY.COM, which reads a keypress without affecting the screen display. It sets ERRORLEVEL to the key's value and uses this value to control batch files.
- KBDFLAGS.COM, which sets ERRORLEVEL to the value of the BIOS keyboard flag area which allows a batch file to detect whether keyboard input is shifted, insert mode is on, and so on.

- KEYSCAN.COM, which determines whether extended keyboard codes are supported; it displays the GETREPLY.COM ERRORLEVEL values for keys.
- VIDTYPE.COM, which determines whether your display is a Monochrome Display Adaptor (MDA), Hercules Graphics Card (HGC), Color Graphics Adaptor (CGA), or Enhanced Graphics Adaptor (EGA) with monochrome, color, or enhanced color screen. It sets the ERRORLEVEL to the video type to control batch files or set environment variables.
- WARMBOOT.COM, COLDBOOT.COM, which allows automatic reset from within batch files instead of asking for the Ctrl-Alt-Del key combination.

GETREPLY.COM allows batch files to change direction in response to a keypress. Conditional tests within the batch files (*IF ERRORLEVEL*) select from the various options, as in Figure A-4. The batch file menus of Appendix C use this program to translate the function keys (F1-F12) into the ERRORLEVEL value.

Figure A-4. YESORNO.BAT is a batch file that uses the ERRORLEVEL from GETREPLY.

```
echo off
:YESORNO
echo Indicate YES or NO by pressing the Y or N key.
echo Press the ESC key to quit.
:QUERY
getreply.com
if errorlevel 27 if not errorlevel 28 goto :END
if errorlevel 78 if not errorlevel 79 goto :NO
if errorlevel 89 if not errorlevel 90 goto :YES
bell.com
goto :QUERY
:YES
echo You pressed Y(es)
goto :YESORNO
:NO
echo You pressed N(o)
goto :YESORNO
:END
exit
```

The sample batch file within Figure A-4 does not specify whether capital letters are required, yet it only checks for the uppercase ASCII values. GETREPLY translates lowercase letters *(a–z)* into their uppercase equivalent (A–Z). Non-ASCII characters, such as function keys (F1–F12), are assigned their keyboard scan code values plus 128. The KEYSCAN.COM program elsewhere in this appendix displays the ERRORLEVEL values GETREPLY generates for the keys. For your convenience, Table A-3 presents commonly-needed values.

Table A-3. Commonly-Used Keypresses and the Resulting ERRORLEVEL Values from GETREPLY.

Keyboard Key	GETREPLY ERRORLEVEL
A–Z	65–90
a–z	65–90
0–9	48–57
F1–F10	187–196
F11–F12	133–134
Esc	27
Enter	13

The GETREPLY program does not echo any keypresses to the screen. It works with both enhanced keyboards (those with 12 function keys) and old-style PC, XT, and AT keyboards having 10 function keys. Program A-4 contains the GETREPLY.SCR script.

Program A-4. GETREPLY.SCR

```
A     0100
JMP   0103
DB    00
XOR   AX,AX
MOV   [0102],AH
MOV   ES,AX
MOV   AX,1200
INT   16
ES:
CMP   AL,[0417]
JNZ   012F
ES:
XOR   BYTE PTR [0417],80
MOV   AX,1200
INT   16
ES:
```

```
CMP   AL,[0417]
JNZ   012F
MOV   AH,10
MOV   [0102],AH
MOV   AX,CS
MOV   ES,AX
MOV   AH,[0102]
INT   16
CMP   AL,00
JNZ   0141
MOV   AL,80
OR    AL,AH
CMP   AL,61
JB    014B
CMP   AL,7A
JA    014B
XOR   AL,20
MOV   AH,4C
INT   21

R     CX
004F
N     GETREPLY.COM
W
Q
```

KBDFLAGS.COM: Keyboard shifts. Once you have beefed up your hard disk system, you may find that it takes a while to run everything within its startup job stream. For instance, you may have set up your STARTUP.BAT (invoked by AUTOEXEC.BAT; see Chapter 4) to perform an integrity check every time you power-up your system. Or it may load a collection of Terminate-Stay-Resident (TSR) pop-up programs.

So what do you do when you want to cut your startup job stream short, but you don't want to risk aborting a disk reorganizing program? You could insert a PAUSE command at pertinent locations within the jobstream. But then you wouldn't be able to get a cup of coffee while your computer initializes itself.

While many programs look for and react to a pending keypress from the keyboard, most don't care whether you merely press one of the shift keys. A few programs, such as screen savers, do. (You may recall that a screen saver is a program that turns off the screen after several minutes without keyboard input.)

A common way to refresh a dimmed display (without affecting the main application waiting there) is to momentarily press the Alt key. The underlying application ignores the keypress, but the screen-dimming TSR knows that a key was pressed. Rather than reading the keyboard buffer, the screen-control program monitors a location in RAM where the BIOS stores the state of the various keyboard shift keys. Table A-4 lists the eight BIOS keyboard shift bit values.

Table A-4. The BIOS KBFLAG Keyboard Shift Bits

KBFLAG	Decimal	Keycap Legend	Persistent
xxxxxxx1	1	Right Shift	No
xxxxxx1x	2	Left Shift	No
xxxxx1xx	4	Ctrl	No
xxxx1xxx	8	Alt	No
xxx1xxxx	16	ScrollLock	Yes
xx1xxxxx	32	NumLock	Yes
x1xxxxxx	64	CapsLock	Yes
1xxxxxxx	128	Ins	Yes

Shift keys are classified as either *momentary* or *persistent*. The Alt, Ctrl, and Shift keys are momentary—they are in effect only while held down. The other shift keys, such as CapsLock or Ins, are persistent—pressing one of these toggles its state on or off.

Persistent keys are good for indicating the occasional exception in a batch file. During the vast majority of normal job stream processing, you take no action. Whenever the exceptional action is called for, you need not wait around, watching intently for the precise time to press the key. You can press it ahead of time, go get that coffee, and come back to find the exception has been handled.

Figure A-5 lists the STARTUP.BAT from one of the systems belonging to the authors. Starting with the nineteenth line, normal processing entails file-packing three 20Mb drives, duplicating their directories, and then loading the application TSR programs listed elsewhere, within APPTSR.BAT.

Figure A-5. A power-up job stream, STARTUP.BAT, that allows an orderly early exit at line 18.

```
echo off
path ..;c:\batch;c:\hdwr;c:\util;c:\dos
verify on
mark !alltsr
mode com1:9600,n,8,1,p
REM use serial laser printer (COM1: with DTR handshake)
mode lpt1:=com1
mode com2:1200,n,8,1,p
echo PRN | print /b:2048 /s:25 /m:5 /u:20
cache-em 512 /p/t
pushdir
mark dostsr
prompt $p$g
set comspec=c:\command.com
set temp=c:\scratch
set tmp=c:\scratch
kbdflags
if errorlevel 128 goto :END
c:\util\vopt c: /n
c:\util\mace\rxbak
c:\util\vopt d: /n
d:
c:\util\mace\rxbak
c:\util\vopt e: /n
e:
c:\util\mace\rxbak
c:
apptsr
:END
exit
```

Each shift key is independent, so more than one BIOS KBFLAG bit may be active at any given time. When you press Ctrl-Alt-Del to reboot your system, KBFLAG looks like this: xxxx11xx, where *x* may be either one or zero. Its decimal value is then 12 (4 + 8), plus whatever other bits are active. The keycap with the largest KBFLAG value is the Insert key. Any KBFLAG value from 128 to 255 would indicate an active Insert key.

Notice that the line 17 of Figure A-5 invokes KBDFLAGS. That program reads the BIOS KBFLAG byte and emits it as the ERRORLEVEL. Referring to Table A-4, you can see that pressing the Ins key prior to that line will result in an ERRORLEVEL of 128 or more. If so, the next line exits the batch file early, skipping the reorganization and TSR-loading.

IF ERRORLEVEL tests that the ERRORLEVEL is equal to or greater than the specified value. Therefore, the Insert key is the easiest persistent key to look for via KBDFLAGS. Program A-5 lists the KBDFLAGS.SCR source code, which creates the KBDFLAGS.COM program.

Program A-5. KBDFLAGS.SCR

```
A     0100
MOV   AX,0040
MOV   DS,AX
MOV   AL,BYTE PTR [0017]
MOV   AH,4C
INT   21

R     CX
000C
N     KBDFLAGS.COM
W
Q
```

KEYSCAN.COM: Keyboard scan codes. With the advent of the enhanced keyboard with 12 function keys, IBM sought to provide a standard keyboard across its various product lines. Unfortunately, the additional keys were not recognized by the earlier PCs, XTs, and ATs. Software which reads the keyboard via the originally-proscribed ROM BIOS keyboard query function (00H) will never detect F11 or F12 keypresses. Conversely, software that uses the new extended query (10H) will not read anything when run with a BIOS that predates the extended-keyboard support.

KEYSCAN.COM is a program that determines whether a given system's BIOS supports the Enhanced Keyboard. It uses either the original function or the extended function, whichever will work, to read the scan codes of keys as they are pressed. Program A-6 lists the DEBUG script to create KEYSCAN.COM.

Program A-6. KEYSCAN.SCR

```
A       0100
JMP     01A0
DB      'KEYSCAN Version 1.4: show keyboard scan codes',0D,0A
DB      'Press any key, <Ctrl-Break> exits.',0D,0A,'$'
DB      00
DB      00,00,00,00,'H, GETREPLY errorlevel '
DB      00,00,00,0D,0A,'$'
DB      'No '
DB      'Enhanced Keyboard BIOS support.',0D,0A,0D,0A,'$'
MOV     DX,0103
MOV     AH,09
INT     21
XOR     AX,AX
MOV     ES,AX
MOV     DX,0179
MOV     AX,1200
INT     16
ES:
CMP     AL,[0417]
JNZ     01DB
ES:
XOR     BYTE PTR [0417],80
MOV     AX,1200
INT     16
ES:
CMP     AL,[0417]
JNZ     01D5
MOV     DX,017C
MOV     AH,10
MOV     [0157],AH
ES:
XOR     BYTE PTR [0417],80
MOV     AH,09
INT     21
MOV     AX,CS
MOV     ES,AX
MOV     AH,[0157]
INT     16
PUSH    AX
MOV     DI,0158
CALL    0215
POP     AX
```

```
OR      AL,AL
JZ      0202
CMP     AL,61
JB      0206
CMP     AL,7A
JA      0206
SUB     AL,20
JMP     0206
NOP
MOV     AL,80
OR      AL,AH
MOV     DI,0173
CALL    0233
MOV     DX,0158
MOV     AH,09
INT     21
JMP     01E3
CLD
MOV     CX,0004
ROL     AX,1
ROL     AX,1
ROL     AX,1
ROL     AX,1
MOV     DX,AX
AND     AL,0F
AND     AL,0F
ADD     AL,90
DAA
ADC     AL,40
DAA
STOSB
MOV     AX,DX
LOOP    0219
RET
MOV     CX,000A
XOR     AH,AH
MOV     DX,AX
MOV     AL,20
MOV     [DI],AL
INC     DI
MOV     [DI],AL
INC     DI
MOV     AX,DX
XOR     DX,DX
```

```
DIV   CX
XCHG  DX,AX
ADD   AL,30
MOV   [DI],AL
DEC   DI
OR    DX,DX
JNZ   0242
RET

R     CX
0153
N     KEYSCAN.COM
W
Q
```

In addition to displaying the hexadecimal scan code and ASCII key value, KEYSCAN also shows what ERRORLEVEL would result from the GETREPLY program sensing the key as its input. You can use KEYSCAN to figure out what ERRORLEVEL values to check in your interactive batch files.

VIDTYPE.COM: Display identification. The IBM PC world is populated with many past and present video standards. Monochrome displays may be driven by the original text-only IBM Monochrome Display Adaptor (MDA). They may use the first monochrome graphics card for the PC, the Hercules Graphics Card (HGC). Or they may run from an Enhanced Graphics Adaptor (EGA). They may even be emulating a Color Graphics Adaptor (CGA), using 16 intensities to simulate the various colors available.

A color screen may be attached to a CGA video board, or perhaps to an EGA. But the EGA might be driving an Enhanced Color Monitor, or even a *variable-scan* device.

Each of these hardware combinations works differently. Each has its own complement of features and limitations. Designing job streams for the lowest common denominator cheats the user of much of the power now available. Why settle for 25 × 80 monochrome text in 8 × 8 character cells, when your hardware can provide 9 × 14 characters, or 60 × 80 screens?

Figure A-6 lists a simple batch file that switches between a color and monochrome display. It is a simple application of the VIDTYPE.COM program.

Figure A-6. SWITCH.BAT moves the active console between a color display and a monochrome screen.

```
echo off
REM switch between monochrome and color video adaptors
cls
vidtype > nul
if errorlevel 4 goto :ISCOLOR
if errorlevel 1 goto :ISMONO
echo Not switching unknown monitor type.
goto :END
:ISMONO
mode co80
goto :END
:ISCOLOR
mode mono
:END
exit
```

The VIDTYPE program determines the active video hardware. By setting the ERRORLEVEL according to Table A-5, the program provides job streams with the ability to take advantage of special hardware features.

Table A-5. The VIDTYPE program uses the ERRORLEVEL to describe the active video hardware.

Video Adaptor	Video Monitor	VIDTYPE ERRORLEVEL
Unknown	Unknown	0
MDA	Monochrome	1
HGC	Monochrome	2
EGA	Monochrome	3
CGA	Color	4
EGA	Color	5
EGA	Enhanced Color	6

The DEBUG script to create the VIDTYPE program appears in Program A-7.

> The following program will not run properly on Compaq computers.

Program A-7. The VIDTYPE.SCR file creates the video hardware detection VIDTYPE.COM program.

```
A     0100
JMP   01A9
DB    '(C) 1987 COMPUTE Publications, Greensboro, N.C.'0D,0A,1A
DB    'EGA$ Enhanced Color$CGA Color$MDA$HGC$
        Monochrome$'
DB    'inactive, plus $Unknown$ video adaptor installed
        (ERRORLEVEL=0)'
DB    0D,0A,'$'
MOV   AX,CS
MOV   DS,AX
MOV   AH,12
MOV   BL,10
INT   10
CMP   BL,10
JZ    0200
MOV   DX,0134
MOV   AH,09
INT   21
MOV   AX,0040
MOV   ES,AX
ES:
MOV   AX,[0087]
TEST  AL,08
JNZ   01F9
MOV   DX,015A
MOV   AL,03
CMP   BH,00
JNZ   023C
MOV   DX,0138
MOV   AL,06
AND   AH,0F
CMP   AH,03
JZ    023C
CMP   AH,06
JZ    023C
CMP   AH,0B
JZ    023C
CMP   AH,0E
JZ    023C
MOV   DX,014B
MOV   AL,05
```

```
JMP  023C
MOV  DX,0166
MOV  AH,09
INT  21
ES:
MOV  AX,[0063]
CMP  AX,03D4
JZ   0215
CMP  AX,03B4
JZ   021C
MOV  DX,0177
MOV  AL,00
JMP  023C
MOV  DX,0148
MOV  AL,04
JMP  023C
MOV  DX,03BA
IN   AL,DX
AND  AL,80
MOV  AH,AL
MOV  CX,1000
IN   AL,DX
AND  AL,80
CMP  AH,AL
JNZ  0237
LOOP 0227
MOV  DX,0152
MOV  AL,01
JMP  023C
MOV  DX,0156
MOV  AL,02
PUSH AX
ADD  [01A4],AL
MOV  AH,09
INT  21
MOV  DX,017F
MOV  AH,09
INT  21
POP  AX
MOV  AH,4C
INT  21

R    CX
0151
```

```
N     VIDTYPE.COM
W
Q
```

WARMBOOT.COM: Automated restart. Have you ever been bugged by a batch job stream that asks you to *Press Ctrl-Alt-Del to Reboot*? You may have asked yourself, "If the process knows enough to ask for the reboot, why doesn't it just reset itself?" If it wants to give the operator the option, it could still ask for a Yes or No response, as in Figure A-7.

Figure A-7. RESET.BAT conditionally resets the computer without requiring the operator to use any special key sequences.

```
echo off
REM ask if system should be reset
echo Do you wish to restart the computer now?
:YESORNO
echo Indicate YES or NO by pressing the Y or N key.
:QUERY
getreply.com
if errorlevel 78 if not errorlevel 79 goto :NO
if errorlevel 89 if not errorlevel 90 goto :YES
bell.com
goto :QUERY
:YES
echo Please wait while the system resets . . .
warmboot
:NO
:END
exit
```

A memory value determines whether the computer will perform its exhaustive power on self test (POST) after a restart. When the two bytes at memory location 0040:0072 are 1234H, the ROM BIOS is tricked into thinking it has already performed the POST. It therefore runs a quick *warmboot*, rather than its more exhaustive initial *coldboot*. Program A-8 is a DEBUG source code to create two restart programs, WARMBOOT.COM and COLDBOOT.COM.

Program A-8. WARMBOOT.SCR and COLDBOOT.SCR

```
A       0100
MOV     AX,0040
MOV     DS,AX
INC     BX
MOV     WORD PTR [0072],1234
CLI
JMP     FFFF:0000

R       CX
0012
N       WARMBOOT.COM
W
A       0105
MOV     WORD PTR [0072],0000

N       COLDBOOT.COM
W
Q
```

Appendix B

Do-It-Yourself Batch Files

Do-It-Yourself Batch Files

MS-DOS offers a handy way to turn an often-repeated sequence of programs or commands into an integrated function. This facility is the BATCH processor built into the MS-DOS Command Line Interpreter, COMMAND.COM. Whenever it encounters a command that is not intrinsic (like DEL or CLS), COMMAND.COM searches first for a program (with a filename extension of .COM or .EXE) and then for a batch file (filename extension of .BAT).

Appendix B covers the whys and wherefores of MS-DOS batch files. Specifically, it explains:

- Redirection and pipes
- Environment variables
- Parameter variables
- Batch file commands
- Design tips and techniques

After reading this appendix, you should understand batch files, such as the ready-to-run examples in Appendix C. Not only will you understand what they do just by looking at them, you'll also know how they do it and why they use specific techniques.

You may skip reading this portion of the book until you need to modify or create your own batch files, but it is recommended that you read it now. When you know the possibilities, you'll tend to work smarter, not harder, with batch files.

Redirection and Pipes

Beginning with Version 2.0, MS-DOS incorporated features similar to those found in Bell Laboratories' UNIX operating system. Because it is multitasking and multiuser, UNIX has been a favorite environment at universities since the mid-1960s. Many operating system programmers have been exposed to UNIX. A significant number of them were nurtured on it.

In addition to the tree directories so essential to hard disk management (see Chapter 4), MS-DOS Version 2.0 instituted five standard UNIX-type files.

- STDIN (standard input) provides the operator input keystrokes. It is usually CON:, the console keyboard, but may be a disk file.
- STDOUT (standard output) receives the program display text. It is usually CON:, the console screen, but may be a disk file.
- STDERR (standard error) receives program error notices. It is assigned to CON:, the console screen.
- STDAUX (standard auxiliary) provides serial input and output on a byte-by-byte basis. It is assigned to the AUX: device.
- STDPRN (standard printer) receives all text sent to the printer. It is assigned to the PRN: device.

MS-DOS directs all data destined for the logical devices CON:, AUX: or PRN: to these standard file *handles*. No reprogramming is necessary in order for applications to use them. However, applications that bypass MS-DOS and communicate directly to the hardware are unable to use the standard files. Unfortunately, most of the fast, popular applications are fast because they bypass the overhead of MS-DOS.

Nature seeks a balance, so it deals in trade-offs. While speed was sacrificed by programs using MS-DOS functions, these programs were made more flexible. Using the MS-DOS system allows a program's stdout data to be written to disk. Its stdin keystrokes can be read from disk. This is known as *redirection*, a UNIX feature incorporated into MS-DOS.

The MS-DOS COMMAND.COM uses the less than symbol (<) to indicate redirected input. It treats the greater-than symbol (>) as a request to redirect output. You may recall from Appendix A that the DEBUG editor can receive its keystrokes from a disk file. Similarly, the DEBUG display can be written to a disk file. For instance, **debug <test.scr >test.lst** is the command line to run DEBUG using the file TEST.SCR for input and with display output directed to the file TEST.LST.

Using double output-redirection symbols (> >) appends the stdout to any existing file, or creates a new file if the named file doesn't exist.

What if one program's disk file stdout were used as another program's disk file stdin? That would form a *pipe*, indicated by the vertical-bar concatenation symbol (|), just as in UNIX. (On the IBM keyboard, the vertical line character has a thin break in the middle.) Programs that participate in pipes are called *filters*. Filter programs generally perform a single, simple function. By combining building-blocks of filters, you can create a job stream that behaves in a very specific manner. For instance, the command **dir | find /v "<" | sort | more** will create a sorted display of the current directory, exclude all subdirectories, and paginate the listing to a single screen at a time.

The DIR command writes directory data to stdout, which MS-DOS saves in a temporary disk file created for the occasion. The FIND filter reads that temporary file as its stdin. FIND writes all the lines, unaltered, to stdout, except the ones containing the less-than symbol (found in subdirectory listings, which contain the word <DIR>). The FIND stdout goes to another temporary file and is read as stdin by the SORT filter. SORT merely puts the lines in sequence, starting from column 1. The sorted list goes via stdout to disk. MS-DOS then supplies the sorted lines to MORE as its stdin. MORE displays a screenful of lines, displays its *—More—* message, and awaits a keypress before displaying the next screenful.

The differences between pipes and redirection are:

- Pipes maintain temporary disk files named, created, and deleted automatically by MS-DOS. The files are not persistent and cannot be appended to. They are always both written and read, created anew with each use and deleted when no longer needed. Their names are unpredictable, so they are difficult for a nonpiped application to manipulate.
- Redirected files are named, created, and deleted under direct control of the user or application programmer. They are as persistent as needed and may be appended to. They may serve as one-way files to increase efficiency. Their names are predictable, so they can be treated as just another data file.

Consider two ways to do the same thing. Say a batch file contains the *DEL *.** command. Upon processing this command, MS-DOS will request verification from the operator (*Are you sure?*) and await a series of keys, terminated by Enter. If the first keypress is Y or *y*, the erasure proceeds. This wholesale file deletion can be automated either by a pipe, as in **echo Y | del *.* > nul** or **del *.* < c: \batch \y.inp > nul**, which operates via redirection of an input file containing the character *Y*.

Both procedures suppress the *Are you sure?* query by redirecting it to the NUL device (see the heading **Suppressing console display** under the Tips and Techniques section of this appendix).

What are the trade-offs with these two approaches? The pipe example is more portable and self-contained, but it requires the creation, writing, reading, and deletion of a temporary file. The redirection is quicker and does not fragment the disk directory because it just reads an existing file, but the input file containing the character *y* must be precreated. Furthermore, its path must be changed if the batch is run on a system with a different directory organization.

Which should you use? That's up to you.

Environment Variables

The MS-DOS environment is a global system memory pool. It holds textual information as a series of ASCII (American Standard Code for Information Interchange) characters, called strings. Each string is delimited by a character of binary zero, a null. This format is called ASCIIZ (zero-terminated ASCII). Every environment string has a name. The names of some environment strings are preordained by MS-DOS, but most can be given any name you choose. A named environment string is commonly referred to as an *environment variable.*

The standard MS-DOS environment variables are *COMSPEC, PATH,* and *PROMPT*. Their functions and accesses are:

- COMSPEC defines the filename and directory from which DOS reloads the command processor after running a batch file or other process. It defaults to the drive, directory, and file from which the system loaded. COMSPEC can be modified by the *SET COMSPEC=* command. Changing COMSPEC is not recommended unless you have a ramdisk and are running MS-DOS 3.0 or later. In that

case, copy COMMAND.COM to the RAM disk and point COMSPEC to the RAM disk version. This will speed up your processing.

- PATH defines the sequence of directories searched when DOS looks for a program or batch file that is not in the current directory. There is no default value. PATH can be modified by the *PATH* or *SET PATH=* commands. The form **.;C:\BATCH;C:\UTIL;C:\DOS** is recommended.
- PROMPT controls the format of the MS-DOS command prompt. When empty, PROMPT behaves as if it were set to *ng* which generates a drive letter and the greater than symbol (>). PROMPT can be modified via the *PROMPT* or *SET PROMPT=* commands. The form *pg*, which generates the current path and the greater than symbol (>), is recommended.

The contents of environment variables may be altered via the SET command. *SET variablename=value* is the format of the SET command, where any spaces between the start of *variablename* and the equal sign (=) are considered part of variablename, and all spaces on the command line after the equal sign are part of *value*.

Enter the command **SET** to view the current values of the environment variables. This displays each variable name and its value, one per line, upon the stdout device. Redirecting this display to a disk file with the command **SET > C:\SCRATCH\OLDSET.DAT** saves all the environment settings in a file named OLDSET.DAT.

An individual environment variable may be redirected as well. For instance, to save the current program search PATH, modify it, run a program, and then restore the original PATH would involve the sequence below:

```
path > c:\batch\oldpath.bat
path c:\ws4
ws c:\batch\oldpath.bat
del c:\batch\oldpath.bat
```

An environment variable is created or modified when it has a value assigned to it, as in **SET TEMP=C:\SCRATCH**.

The variable is deleted when it is made empty, as in **SET TEMP=**, when the command line ends immediately after the equal sign.

An environment variable is limited in size to the length of the command line that defines it. SET uses up four of the 127 command-line characters, and the equal sign takes another. The maximum length of an environment variable name and its value, combined, is 122 characters.

Environment variables are directly accessible by batch files. An unnested batch file (not invoked via COMMAND /C) can set the system-wide variables. Nested batch files only affect the variables for the secondary command processor. Whether nested or not, batch commands can use environment variables as parameters.

The batch processor considers a character sequence between two percent signs (% %) to be an environment variable name. DOS then substitutes the string value of the variable for its name in the command line. If no matching variable name is found, the two percent signs and the intervening text disappears. For instance, the batch command **DIR %BOGUS%** becomes **DIR** if the environment variable BOGUS doesn't exist.

The amount of memory set aside for the environment area is configurable. The default area is 160 bytes. A 992-byte area is the maximum allowed under MS-DOS versions earlier than 3.2. With MS-DOS 3.2, the limit jumped to 32K.

A SHELL statement parameter within the CONFIG.SYS allows defining the environment space for MS-DOS versions 3 and higher. Versions 3.0 and 3.1 used memory allocation in 16-byte paragraphs. Starting with Version 3.2, the same parameter was measured in individual bytes. Specifying too many paragraphs forces MS-DOS to allocate the maximum space under MS-DOS Versions 3.0 and 3.1. So the SHELL statement **shell=c:\command.com /e:992 /p** nets the same 992-byte area in all 3.*x* MS-DOS systems.

MS-DOS 2.0 and 2.1 users should be aware that the Microsoft C compiler package includes the SETENV program which will perform an officially-sanctioned *patch* to the COMMAND.COM for Version 2.*x* MS-DOS that extends the environment area to 992 bytes.

Parameter Variables

Batch files are invoked when the command line interpreter, COMMAND.COM, analyzes a command line. COMMAND.COM treats spaces as separators, as well as semicolons (;) and commas (,). Each series of command line characters between separators is construed to be a parameter.

The batch processor recognizes special replaceable parameter variables. These consist of a percent sign (%) followed immediately by a single decimal digit. When analyzing a command line, the batch processor replaces instances of parameter variables with corresponding command line parameters. For instance, if the batch file VERIFY.BAT consisted of **chkdsk /v %1** and the invoking command line were **verify c:**, then the batch processor would substitute **c:** for **%1** and **chkdsk /v c:** would be run.

Unless renamed by the batch SHIFT command (described later in this appendix), the first command-line parameter is *bound to* (substituted for) %0, the second is %1, and so on. The name of the batch file being invoked is always %0. When displaying usage instructions from within a batch file, use the variable rather than *hard-coding* the batch file name into the examples. Then renaming the batch file does not entail any editing. Referring to the preceeding batch segment, **echo To verify drive C:, type "%0 C:" and include the colon** might be the way its usage sample would be written.

Although only ten command-line parameters are accessible at a time (%0 through %9), many more may be entered. A command line is limited to 127 characters in length, so 64 one-character parameters are possible. The SHIFT command makes any additional parameters available, one at a time.

Batch File Commands

MS-DOS batch files may be as simple as text files that contain the MS-DOS commands which you use to perform various tasks. For instance, Figure B-1 lists a one-line batch file that moves from the current directory to the parent directory, one level closer to the root directory (these terms are fully defined in Chapter 4). How useful is this batch file? Is it easier to remember and type CD . . or UP when you want to move up a directory level? The batch file is mnemonic and requires half as many keystrokes.

Figure B-1. UP.BAT moves to the parent directory.

```
cd ..
```

As discussed in Chapter 5, batch files may combine a series of processing steps into integrated job streams. By putting all the related programs into a file that is invoked with a single command, you insure that steps always happen in the correct sequence and that no important steps are left out.

In many offices, laser printers are shared between two or more computer systems. Although Hewlett-Packard warns against it, many LaserJet printers communicate via serial cables attached to A-B switches. Figure B-2 is USELASER.BAT, a batch file that reminds the operator to set the switch, then verify that the printer is on. Next, it insures that the communications line uses the proper protocol (9600 baud, no parity, 8 data bits, 1 stop bit, DTR flow-control). Then it reassigns the logical printer device (PRN) to the serial (COM1) physical device—all in response to the command USELASER.

Figure B-2. USELASER.BAT assigns the printer to the serial laser printer.

```
echo off
rem assign PRN to serial laser printer
echo Ensure the laser printer is ready and the A-B switch set.
pause
mode com1:9600,n,8,1,p
mode lpt1:=com1
cls
echo The laser printer is now selected as your printer.
```

Even when people have access to a laser printer, they often use a dedicated parallel dot-matrix printer. Figure B-3 shows a simple batch file, USEDRAFT.BAT, to reassign the logical printer device (PRN) to the parallel (LPT1) physical device. USEDRAFT is the complement to the USELASER job stream.

Figure B-3. USEDRAFT.BAT assigns the printer to the parallel dot-matrix device.

```
echo off
rem assign PRN to parallel dot-matrix printer
mode lpt1:
cls
echo The draft printer is now selected as your printer.
```

In addition to MS-DOS commands you can type yourself at the command-line prompt, the two preceeding batch files contain batch-only commands. ECHO and PAUSE are commands that are understood by COMMAND.COM only while it is running a batch file. Adding batch commands extends the usefulness of batch files.

MS-DOS provides a rudimentary, yet powerful, batch processing language. The batch commands, in the sequence we examine them, are:

- ECHO displays batchfile text upon the console.
- PAUSE requests and waits for a keypress.
- REM contains a nonexecuting comment.
- IF condition-testing lets batch files vary their processing in response to external situations.
- GOTO changes the order of processing within a batch file.
- FOR repeats a process for a set of files or commands.
- SHIFT renumbers replaceable batch parameters.
- CALL allows *nesting* of batch files.

ECHO: Display batch text. The ECHO command controls whether MS-DOS displays batch file commands upon the console screen. Normally, COMMAND.COM sends an echo of what it reads from the batch file to the display. This is the effect of the ECHO ON batch command.

However, another ECHO command suppresses this onscreen echo. To reduce the onscreen clutter, many batch files begin with the ECHO OFF command.

Under MS-DOS Version 3.2 and earlier, the ECHO OFF command is itself displayed upon the screen. Beginning with Version 3.3, batch file lines that begin with a commercial at sign (@) are not displayed. Therefore, for MS-DOS 3.3 or later, begin batch files with @ECHO OFF.

ECHO by itself on the command line generates a status report, such as *Echo is on*. Any other form of the ECHO command sends the remainder of its command line to the display. This facility is useful when the now-uncluttered screen provides no useful indication of what is going on. Use ECHO to give brief instructions to the operator; use TYPE or COPY for more than a few lines. The command ECHO+ sends a blank line, as in the DOWN.BAT procedure in Figure B-4.

Figure B-4. DOWN.BAT uses ECHO+ to display a blank line to separate the directory list from the prompt.

```
echo off
REM move to child directory (down one level)
if not "%1"=="" goto :DIR
dir | find "<" | sort | more
echo+
echo Move to what subdirectory?
goto :END
:DIR
cd %1
:END
exit
```

PAUSE: Wait for operator response. The PAUSE command halts a batch file until the operator presses a key. Therefore, PAUSE can pace the batch file, relying upon the operator to indicate when it is time to proceed. It is often used in conjunction with the ECHO command, as in Figure B-5.

Figure B-5. The PAUSE batch command waits for the operator to press a key before proceeding.

```
echo off
rem Print WordStar file on serial printer using stationery
echo Manually feed a letterhead into the printer.
echo+
pause
pslist -w -s %1 com1:
```

The PAUSE command prompts the operator to *Strike a key to continue*. The program doesn't sense which key was pressed, unless it is the Ctrl-Alt-Del reboot combination. This restart triad need not be entered in response to a PAUSE command, however. A reboot can be invoked at any time.

If your batch process must distinguish between various possible keyboard responses, use the GETREPLY.COM program from Appendix A instead.

REM: Documenting job streams. The self-evident batch procedure you write today could become the obscure incantation of tomorrow. Relatively little software support effort goes into developing new software. Most of it is maintaining existing programs and procedures. Job streams are no different. Make yours easy to maintain by making them self-documenting.

MS-DOS supports batch file comments. The REM statement causes COMMAND.COM to ignore everything up to the end of the line. The end of the line is either a carriage return (13), a line feed (10), a pipe command (|), or a redirection symbol (< or >).

After suppressing echo, begin your batch files with a statement of their purpose, as in Figure B-6.

Figure B-6. KERMIT.BAT self-documents both its purpose and its restrictions.

```
echo off
rem public-domain Kermit telecommunications
echo Verify that the modem is turned on and connected.
echo+
pause
cls
rem Kermit ver. 2.26 only works with files in the current directory
c:\comm\kermit\mskermit.exe
```

Notice that Figure B-6 contains two comments. The second remark, towards the end of the file, documents an application's idiosyncratic behavior. In this way, viewing an infrequently run batch file can spare you wasted effort.

Use remarks to note why you do things a certain way. When a later version of the software overcomes a limitation, the notes can remind you to use a better method.

IF: Evaluating conditions. The IF batch command lets a batch file conditionally execute a command. With this facility, a batch file can include or exclude processing based upon one of three conditions:

- ERRORLEVEL being equal to or greater than a threshold
- Two text strings being identical
- A file existing

The remainder of the command line after the *IF condition* is performed only if the condition is true. A complementary format, *IF NOT condition* performs the tail of the command line only when the tested condition is false.

Beginning with MS-DOS Version 2.0, programs have the ability to tell the program that started them how well they processed. This is communicated in an eight-bit *return code*, allowing a value from 0 to 255. When the COMMAND.COM program runs a program, this return code is the system ERRORLEVEL.

Few programs bother to set the ERRORLEVEL. Those that do generally adhere to only one rule: ERRORLEVEL 0 means everything worked. Once set, ERRORLEVEL retains its value until a program specifically changes it.

The IF command can test for a threshold value in the ERRORLEVEL. The syntax is *IF ERRORLEVEL value*, where *value* ranges from 0–255. An IF ERRORLEVEL is true when the ERRORLEVEL is equal to or greater than the threshold value in the test. For instance, the batch file command **IF ERRORLEVEL 1 ECHO ERROR! PROGRAM ENDED ABNORMALLY** displays a message only if a program has ended with an ERRORLEVEL equal to or greater than 1.

IF NOT ERRORLEVEL 1 ECHO SUCCESSFUL COMPLETION uses the *IF NOT condition* syntax to display a message only if the preceeding programs had no errors.

An IF ERRORLEVEL test does not alter the ERRORLEVEL setting. Therefore, multiple ERRORLEVEL tests may be performed.

String comparison is very useful in batch files. A string, for batch file purposes, is a sequence of text characters with no separators, such as spaces, commas (,), equal-signs (=), or semicolons (;). The string comparison syntax is *IF string1 == string2* with optional spaces on either end of the double-equality sign (= =), but none embedded within it.

A command such as **IF ABC==ABC** will always be true and is therefore pointless. Variability comes from variables. The variables to use in these comparisons are command-line parameters named %0 through %9 within the batch file. Environment variables are named as their environment-name (used with the SET command) between two percent signs (%). For instance, the current command-line prompt is controlled by the PROMPT or SET PROMPT= command. The PROMPT string value is therefore accessible within a batch file as **%PROMPT%**.

MS-DOS replaces the variable names in the command line with their actual values before checking the strings for equality. If no variable exists or it has no value, the variable name disappears. This can cause syntax errors, because **%2==ABC** becomes **==ABC** if fewer than two command-line parameter variables are entered. To prevent a one-sided equality test, at least one extra character should be inserted either before or after the string value.

For instance, the code fragment **IF %1X==SECRETAGENTX MENU** would run the MENU program only if the password SECRETAGENT were used.

A common method of verifying that enough parameters are entered is to check for an empty last parameter. **IF "%2"=="" EXIT** insures that at least two parameter variables were specified, and it aborts if the second variable is empty.

The *IF EXIST filename* syntax verifies the presence of the specified file. A batch job stream can therefore trigger an action based on the existence of certain files. Suppose you had a job stream that included a first-time setup program, INSTALL, which saved the user's configuration preferences in a file called CONFIG.DAT. The job stream batch file below would then run the INSTALL program only if no configuration data file already existed. It would run its application menu program, APPMENU, after the configuration data had been created:

```
if not exist config.dat install
if exist config.dat appmenu
```

Multiple conditions may nest within another on a single command line. To take an action only if the ERRORLEVEL is a certain specific value requires two tests. The first test looks for the ERRORLEVEL being equal to or greater than the target value. The second test (on the same line) then proceeds only if the ERRORLEVEL is not greater than or equal to a value one larger than the threshold. For instance, **IF ERRORLEVEL 8 IF NOT ERRORLEVEL 9 DEL PAYROLL.TMP** deletes PAYROLL.TMP only when the ERRORLEVEL is 8.

A typical action that results from IF tests is to alter the processing sequence. This alteration is accomplished via the GOTO command, discussed next.

GOTO: Altering batch sequence. Using the IF command, a batch file may conditionally perform one command. If that command is a batch GOTO, the single conditional test need only be made once. Because the GOTO alters the processing sequence of the batch file, it effectively applies the test to an entire block of commands. A GOTO command causes COMMAND.COM to continue processing from an arbitrary point in the batch files, rather than reading and processing the next line.

The syntax of the GOTO command is *GOTO label.* A *label* is a line consisting of a string that begins with a colon. While the colon is mandatory for the label definition itself, it is optional for the GOTO command. Referring to Figure B-7, the OVER.BAT batch file contains two labels, DIR and END.

Figure B-7. OVER.BAT is a directory-traversal job stream containing two labels.

```
echo off
REM move to a sibling (same level) directory
if not "%1"=="" goto :DIR
dir .. | find "<" | sort | more
echo+
echo Move to what directory?
goto :END
:DIR
cd ..\%1
:END
```

OVER.BAT runs starting at label DIR if a parameter is given, otherwise it displays a list of possible values and exits via label END. Thus, a GOTO extends a single IF test to encompass multiple batch file lines.

Not all GOTO commands are conditional, however. Near the end of the batch file in Figure B-7 is an unconditional *branch,* or change in processing sequence, that always skips from the message display to the end of the job stream.

FOR: Processing sets. While many MS-DOS commands can use ambiguous wildcard filenames as parameters, others cannot. The FOR command works with members of a set, one at a time. *FOR variable IN (set) DO process* is the syntax of the FOR command. The *variable* is either one or two percent signs followed by a single alphabetic character, such as %%f. Use a double-percent within a batch file. (The single-percent should be used with a FOR command entered in direct mode from the MS-DOS command line.)

The *set* is a space-separated list of items (such as *C: D: E:*). It may contain only a single item, and that item may be an ambiguous filename such as *.DOC. The *process* is a command line, just as with the IF command.

The FOR command is not exclusively a batch command, but it's used rarely outside of batch files.

If the *set* is an ambiguous filename, then the *variable* is assigned the name of every matching file, once per process. To illustrate, suppose directory C: \TEXT \ contained the three files A.TXT, B.TXT, and C.TXT. The batch command to replace them with their respective copies on floppy disk drive A would be:

```
cd c:\text
c:
for %%f in (*.txt) do copy a:%%f
```

This generates %%f substitutions of A.TXT, B.TXT, and C.TXT. Additional text files from drive A are not copied. Sometimes you may need to run multiple commands on a wildcard set of files. Consider a different way of specifying the set. If the commands can all handle the wildcard filename directly, you can speed processing if you define the set as the list of commands. For instance, both COPY and DEL can handle ambiguous filenames. The batch command **FOR %%C IN (COPY DEL) DO %%C A:*.TXT** copies all text files from floppy disk drive A to the current directory and then deletes them from the floppy disk.

FOR commands can be nested, but only by calling a second COMMAND.COM. This is accomplished by inserting *COMMAND/C* ahead of the *process*. For instance, the batch command to copy the aforementioned files and at the same time remove only the matching floppy versions is:

```
cd c:\text
c:
for %%f in (*.txt) do command/c for %%c in (copy del) do %%c a:%%f
```

If the set specification includes any drive or path specification, the entire specification is bound to the variable. Referring back to the text file example, it was necessary to first log into the C: \TEXT \ directory. Otherwise the set specification would have been C: \TEXT *.TXT. This would result in %%f bindings of C: \TEXT \A.TXT and so on, causing an attempt to copy A:C: \TEXT \A.TXT. You can see that the FOR command requires careful planning and testing.

SHIFT: Renaming parameters. The MS-DOS batch processor within COMMAND.COM refers to the first ten parameter variables entered into the command line as %0 through %9. COMMAND.COM replaces these parameter names with their textual values prior to evaluating conditions and executing the command line. But what if you need 11 parameters? Or 40? Or what if you usually need three, but occasionally eight?

The SHIFT command renumbers the replaceable parameter variables. Every time the SHIFT command runs, the value bound to %1 becomes bound instead to %0, %2 becomes %1, and so on. The 10th parameter becomes %9. The first parameter disappears forever. There is no corresponding UNSHIFT command. Any variables that must be retained after the SHIFT should be SET to an environment variable prior to shifting, as in the line after the LOGIN label of Figure B-8.

Figure B-8. MOVE.BAT uses the SHIFT command to allow it to copy and delete a variable number of files.

```
echo off
REM moves and deletes variable number of files from various sources
if not "%2"=="" goto :CHKDIR
echo This procedure copies and then deletes files from various
   sources.
echo The first parameter is the destination drive and directory. All the
echo other parameters are files to be moved (copied and then deleted).
echo Both specific and ambiguous filenames are supported.
goto :ERREND
:CHKDIR
if not exist %1 goto :LOGIN
echo The first parameter must be the destination drive and path.
goto :ERREND
:LOGIN
set var1=%1
:LOOP
shift
echo on
copy %1 %var1%
echo off
echo Y | del %1 > nul
if not "%2"=="" goto :LOOP
echo+
echo Process completed.
```

```
goto :END
:ERREND
echo+
echo Process aborted.
:END
set var1=
exit
```

The SHIFT command is handy for those batch jobs which require a variable number of parameters each time they are run. For instance, Figure B-8 contains a batch procedure to copy and delete any number of files from a variety of sources.

The SHIFT command is a necessity for batch files that require more than ten replaceable parameter variables. The job stream within Figure B-9 is ambitious. It allows up to 40 parameters. That batch writes a DEBUG input script (see Appendix A) and then invokes the DEBUG editor to create a program for sending setup codes to the printer.

Figure B-9. PRNSETUP.BAT uses the SHIFT command to write as many as forty parameters into a program file.

```
echo off
REM Creates COM file to send codes to the printer
if not "%2"=="" goto :CHKUPDT
echo This procedure creates a printer setup program. It needs as its
echo first parameter the name of the program being created. It also
echo requires from 1-40 hexadecimal (00-FF) control codes to
echo send to the printer. For instance, to create a program named
echo C:\UTIL\TOF.COM that advances to top-of-form (0C in hex),
   you'd
echo type "%0 c:\util\tof 0c" and you would run the resulting
echo program by typing "c:\util\tof"
goto :END
:CHKUPDT
if not exist %1.COM goto :MAKEIT
echo Program "%1" already exists. Do you wish to overwrite it?
echo Indicate YES or NO by pressing the Y or N key.
set adr1=MAKEIT
goto :YESORNO
:MAKEIT
set var1=%1
echo Processing, please wait...
echo a 0100 > c:\batch\prnsetup.inp
```

```
echo jmp 0107 >> c:\batch\prnsetup.inp
echo mov ah,5 >> c:\batch\prnsetup.inp
echo int 21 >> c:\batch\prnsetup.inp
echo ret >> c:\batch\prnsetup.inp
:LOOP
shift
echo mov dl,%1 >> c:\batch\prnsetup.inp
echo call 0102 >> c:\batch\prnsetup.inp
if not "%2"=="" goto :LOOP
REM end program after last code
echo int 20 >> c:\batch\prnsetup.inp
echo+ >> c:\batch\prnsetup.inp
REM end program after 40th code in case too many entered
echo a 01cf >> c:\batch\prnsetup.inp
echo int 20 >> c:\batch\prnsetup.inp echo+ >>
   c:\batch\prnsetup.inp
echo r cx >> c:\batch\prnsetup.inp
echo 00d1 >> c:\batch\prnsetup.inp
echo n %var1%.com >> c:\batch\prnsetup.inp
echo w >> c:\batch\prnsetup.inp
echo q >> c:\batch\prnsetup.inp
command /c debug < c:\batch\prnsetup.inp > nul
del c:\batch\prnsetup.inp
echo Do you wish to test program "%var1%" by running it now?
set adr1=TESTIT
:YESORNO
echo Indicate YES or NO by pressing the Y or N key.
:QUERY
getreply
if errorlevel 27 if not errorlevel 28 goto :END
if errorlevel 78 if not errorlevel 79 goto :END
if errorlevel 89 if not errorlevel 90 goto :%adr1%
beep
goto :QUERY
:TESTIT
echo on
%var1%
echo off
:END
set var1=
set adr1=
exit
```

CALL: Sharing the environment. The CALL subcommand (available only in MS-DOS versions 3.3 and later) allows your batch files to call other batch files, or even themselves. In the event that they call themselves (known as *recursion*), be sure that they terminate at some point, or you will cause the computer to crash. The power of the CALL subcommand is that it allows your batch files to treat other batch files as if they were commands to be called conditionally. CALL is not to be used with piping or redirection.

The syntax for CALL is simple. To call a batch file named BATCH.FIL from within another batch file, enter the line **CALL BATCH.FIL**, and if the file being called were on another device or in another directory, simply specify the path as in **CALL C:\SCRATCH\BATCH.FIL.**

Programmers might also be interested to know the following information: To load programs (.COM or .EXE), COMMAND.COM first allocates all remaining memory. Then it copies its own version of the environment variables into the program's memory block. At the end of this copy, COMMAND.COM places a Program Segment Prefix (PSP). The address of the local copy of the environment is placed within the PSP at offset 002CH. COMMAND.COM loads the program (at PSP + 0100H) and then passes control to it.

Batch files do not create a copy of the environment; they affect the variables within the memory space of COMMAND.COM. If a batch file is running within a secondary processor (loaded via COMMAND /C), then only the secondary copy of the environment is affected.

Any environment variables changed by the program are located within the local copy of the environment. When the program ends, its changes disappear with its copy of the environment.

When a program or command processor is loaded via CALL, the environment address (at PSP + 002CH) points to the same environment as the current COMMAND.COM.

Design Tips and Techniques

This appendix ends with some recommendations:

- Providing operator feedback
- Displaying screens of text
- Suppressing console display
- Making and using zero-length files
- Graceful exits

Feedback. People don't trust computers. They distrust taciturn computers the most. Not knowing what to do when faced with an unmoving, blinking cursor turns bewilderment into outright hostility.

Let the user know that something is happening. Many a file has been corrupted because an impatient or worried operator reset the computer just in case something was wrong. The line **ECHO PROCESSING WILL TAKE ABOUT TEN MINUTES. PLEASE WAIT . . .** is an excellent use of the ECHO command to indicate that lengthy processes have begun—putting the user's mind at rest.

Anyone could find a better way to spend his time than watching an unchanging computer screen. Don't force an operator to wait, staring, until the screen silently asks for the next action in a lengthy process.

Use an audible tone to signal when a long process is over. Appendix A shows how to make the small BELL.COM program. Use it or something equivalent to notify the user of a change in status or the need for keyboard entry. But don't overuse it. A constantly-beeping job stream annoys everyone. Before you release a job stream, run it a few times to get a feel for the timing.

Use a line like **ECHO FILES ARE NOW RESTORED. SAVE THE BACKUP MEDIA** to inform the user when the batch file is finished. Whenever someone has to guess what the system is doing, count on them guessing wrong.

When things go awry, say so. Require a user response to insure that he knows there is a problem. Let him know how severe the problem is, and what to do about it:

```
bell
echo File restore was not completed! You have a bad backup!
echo Files are bad. Do not continue without a good restore.
echo Label this bad backup and restore with the prior set.
echo You must rerun everything since the last good backup.
pause
```

Displaying text files. ECHO is too slow for sending many lines of text to the screen. Use TYPE or COPY to send a prewritten text file. This also keeps the batch file smaller and speeds its processing.

Create the text file with an editor program. Use the EDLIN program or a word processor in *nondocument mode* to create normal ASCII text. To enter the IBM extended characters, such as the line-drawing set for making boxes and borders, press and hold down the Alt key, press the three digits on the numeric pad that are the decimal value of the desired character, then release the Alt key.

If the text does not contain any ASCII end-of-file characters (1AH, the small right arrow in the IBM extended character set), you can TYPE the text file onto the screen. An even faster method is to COPY the text as a binary file to the console display. However, you must also suppress the *1 File(s) Copied* message by redirecting standard output (stdout) to the null device as in **COPY TEXTFILE.MNU/B CON: > NUL**.

Suppressing console display. To keep an uncluttered screen, you turn off command-line echo within your batch files. To automate processing, you rely upon pipes or redirected keystroke files. But some programs insist upon sending messages to the console. And since the keystrokes they elicit are already being supplied, the user can become totally confused.

A simple and often effective way to eliminate screen clutter is by redirecting stdout to the NUL device, sort of a bottomless "bit bucket." This approach works if the program uses stdout for the offending messages, and the screen display is not needed for a pipe. When first invoked, the PRINT spooler program asks what device is being printed to. The batch command **ECHO PRN | PRINT > NUL** suppresses the question and provides the answer without printing anything on the screen.

Sometimes the stdout stream is used in a pipe, but screen messages still appear. Or perhaps redirecting stdout to NUL does not help. This indicates that the stdout file is not the problem.

Perhaps the stderr file is involved. It cannot be redirected without customizing COMMAND.COM, so redirection is not a viable solution. However, we can reassign the console via the standard CTTY command. By changing the console to the NUL device, all screen activity going through MS-DOS disappears. This is accomplished with the **CTTY NUL** command.

Once the offending program completes, another CTTY command reattaches to the CON console. The command **CTTY CON** reestablishes the console.

However, this detach/reattach strategy is risky. If your console is detached, you have no way of knowing when something goes wrong. Short of a power-off or hardware reset, you cannot interact with the system until the batch job stream reattaches the console.

Zero-length files. Zero-length files are useful. They reside only within the directory. They don't take up any file clusters. Redirecting a batch file REMARK in the example **REM > EMPTY.FIL** creates a zero-length file.

How do you take advantage of a zero-length file? One way is to use the IF EXIST condition within a batch file. For instance, when creating a directory, create an empty file with the directory name. Then you can use IF EXIST to verify that the directory is present, as in Figure B-10.

Figure B-10. NEWCHILD.BAT tries to create a subdirectory only if it does not already exist.

```
echo off
REM give birth to (create and label) new child directory
if not "%1"=="" goto :LOOK
echo This procedure creates and labels a new child directory.
echo You supply just the subdirectory name, such as "%0 LOTUS".
echo The current drive and directory must be the parent directory.
goto :END
:LOOK
if exist .\%1\%1 goto :OLD
md %1
rem > .\%1\%1
if exist .\%1\%1 goto :LOG
echo Unable to create and label child directory "%1"
goto :END
:OLD
echo Child directory "%1" already exists
:LOG
cd %1dir
:END
exit
```

Another use for a zero-length file is when you want to append files but may not always want to create a new destination file. In that case, instead of copying the first source file to create the destination file, begin with a zero-length file, then append the other files to it. Figure B-11 presents COMBINE.BAT, which uses this technique.

Figure B-11. COMBINE.BAT creates a zero-length file so it can always append to an existing destination file.

```
echo off
REM Combines 3rd thru nth files into 1st one
if not "%3"=="" goto :MAKEDEST
echo This procedure appends files. It requires at least three
   parameters. The
echo first parameter is the destination file, which may not already
   exist. It
echo may not be an ambiguous filename; that is, it must not contain
   any wild-
echo card characters (? or *). The second parameter indicates whether
echo the files are binary (B) or ASCII text (A). All remaining
   parameters
echo are the files to be appended. These filenames may include
   ambiguous wild-
echo card file names. For instance, to add all .BAS files and
   SAMPLE.DOC to
echo BIGFILE, enter the command
echo    %0 BIGFILE A *.BAS SAMPLE.DOC
goto :ERREND
:MAKEDEST
if exist %1 goto :APPEND
rem create empty destination by redirecting remark to it
rem > %1
if exist %1 goto :APPEND
echo The first parameter, "%1," must be an unambiguous destination
   file.
goto :ERREND
:APPEND
if "%2"=="A" goto :SETUP
if "%2"=="a" goto :SETUP
if "%2"=="B" goto :SETUP
if "%2"=="b" goto :SETUP
echo The second parameter, "%2," must be A for ASCII or B for
binary.
goto :ERREND
:SETUP
set var1=%1/%2
echo Destination file is "%1"
:LOOP
echo Appending "%3"
copy %var1% + %3 > nul
```

```
shift
if not "%3"=="" goto :LOOP
set var1=
echo+
echo Process completed.
goto :END
:ERREND
echo+
echo Process aborted.
:END
exit
```

Graceful exits. The EXIT command terminates a secondary command processor. It provides a fast way to end a nested batch job. Rather than going to a label at the end of the job stream, the nested batch file can immediately EXIT.

The EXIT command does nothing within the primary command processor, however. Many batch file authors rely upon the batch processor reaching the end of the batch file to terminate the process. This works well enough at first, while you are producing one batch file at a time.

However, there is at least one benefit of having a library of batch routines. Rather than redesigning a job stream from scratch, your library will allow you to copy a line from here, alter a routine from there.

The danger of this piecemeal assembly is that individual batch files that exited by default can be grafted together, turning what used to be a harmless exit path into a destructive monster.

Writing the EXIT command at the end of all your batch files has no effect when you use them by themselves or when they are nested. But when you become a batch file surgeon, visually inspecting your handiwork, an EXIT command sandwiched in the midst of a process should alert you that your scalpel slipped; the patient needs looking-after.

Appendix C

Ready-to-Use Batch Files

Ready-to-Use Batch Files

This appendix presents batch files not listed elsewhere within this book. This appendix is divided into three major sections.

- Batch Menu System presents the DEBUG input scripts and batch files to create a menu-driven environment.
- Application Job Streams lists batch files to run various commercial software packages.
- System Support Job Streams contains various utility job streams to perform system-wide support functions.

Many of the following batch files make references to specific disks and directories. They have all been tested on systems with multiple hard disks. All drive references are to a high-density (1.2Mb) floppy disk drive A, a low-density (360K) floppy disk drive B:, and a single hard disk drive C.

The job streams reference and require these directories:

- C: \BATCH: all menu and system job streams.
- C: \UTIL: all utility support programs, including those from Appendix A. If your system names this directory something else (like C: \BIN \USER), you need to modify a few of the batch files.
- C: \SCRATCH: temporary files.

Although not required, the following directories are assumed to exist on your system:

- C: \DOS: contains only the standard MS-DOS or PC-DOS programs; some systems name this directory C: \BIN;
- C: \HDWR: hardware-specific utilities, such as screen control for video adaptors which emulate various cards, or non-DOS device drivers loaded by CONFIG.SYS.

Given the preceding directory structure, we recommend that the program search path be *.;C:\BATCH;C:\HDWR;C:\UTIL;C:\DOS.*

When creating the job streams listed throughout this book, put all the .BAT batch files and their associated .INP input files into the C: \BATCH directory. Whenever you create a DEBUG input script (filename extension .SCR) from Appendix A or Appendix C, place it in directory C: \SCRATCH.

The following two short batch files run the DEBUG input scripts. Figure C-1 lists BUILD.BAT, which simply redirects a script into DEBUG. Notice that it activates console echo while DEBUG runs. This lets you spot any errors caused by miskeyed scripts.

Figure C-1. BUILD.BAT redirects an input script into the DEBUG.EXE editor.

```
echo off
REM runs DEBUG scripts
if not "%1"=="" goto :DOIT
echo Specify a DEBUG script, such as "%0 sample.scr"
goto :END
:DOIT
echo on
debug< %1
echo off
:END
exit
```

SCRIPTS.BAT, in Figure C-2, automates running the DEBUG scripts. It places the results in the current directory, unless the scripts themselves specify a drive or pathname. To run the scripts within this book, enter the commands

```
c:
cd c: \util
scripts c: \scratch \
```

Figure C-2. SCRIPTS.BAT uses all the DEBUG scripts in the specified directory.

```
echo off
REM runs DEBUG scripts
if not "%1"=="" goto :DOIT
echo This procedure uses the DEBUG program to translate text scripts
(.SCR)
echo into programs. You must enter the complete drive identifier and
  optional
echo directory path (ending with \) where the .SCR files are located.
```

```
  The
echo new programs will be placed into this directory:
cd
echo+
echo To run scripts that are on a floppy disk, you might enter the
  command
echo %0 A:
echo+
echo To use scripts located within another hard disk subdirectory, you
echo could, for example, type %0 C:\SCRATCH\
goto :END
:DOIT
for %%f in (%1*.scr) do command/c build.bat %%f
:END
exit
```

If you want to run your system using menus, continue to the "Batch Menu System" section. You may elect to skip it if you're already using an operating environment, such as Quarterdeck's DESQView, Digital Research's GEM, or Microsoft *Windows.* The Application Job Streams are batch files to invoke various commercial software packages, both from the command line and from within a menu system. The System Support Job Streams are command-line oriented.

Batch Menu System

The MS-DOS command line is not the ideal system environment for everyone. Nor are pointer-driven interfaces like GEM or *Windows.* Perhaps a menu system is the answer.

A menu is merely a listing of choices. The menu system presented here uses text files and batch job streams, plus the following programs from Appendix A:

- COLOR.COM controls screen color; optional if your system is monochome and you prefer the light letters upon a dark backgound provided by the CLS command.
- CURSOR.COM controls screen cursor size and location; required for erasing error messages at bottom of menu screen.
- GETREPLY.COM reads operator keypresses; required to determine which menu option was selected.

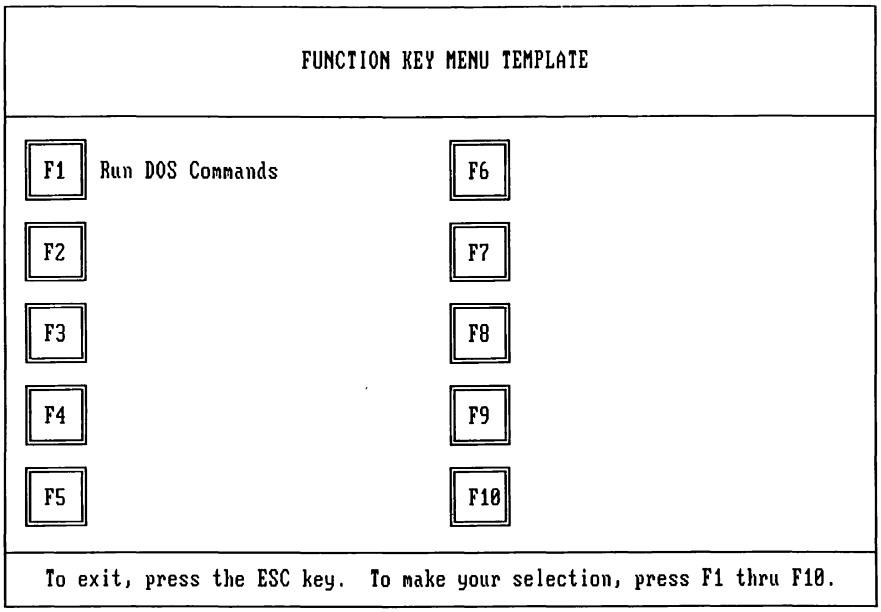
Prototype menu screens. The Batch Menu System uses function keys to select its menu options. The world of PCs is populated with keyboards having either 10 or 12 function keys. This menu system therefore provides two different screen layouts, FKEY10.MNU and FKEY12.MNU.

These screen layouts are generated as text files, using redirected input to the DEBUG editor. To develop a custom menu, you first copy the appropriate prototype screen. This insures a consistent look, and spares you the effort of using the line-drawing characters to redraw each menu separately. Then you employ an editor program or a modified version of one of the enclosed DEBUG scripts to insert your own selections into the copy. A later section of this appendix shows how to customize menus.

Each screen display file consists of 23 lines of 79 characters. Each line ends with both a carriage return (0DH) and a line feed (0AH) character. The lines are one column narrower than the screen to prevent line-wrap. They are two lines shorter than the screen to allow messages at the bottom of a menu display.

Figure C-3 displays the prototype menu screen for systems with an old-style PC, XT, or AT keyboard having ten function keys

Figure C-3. FKEY10.MNU contains the prototype screen display for 10-option menus.



numbered 1–10. This menu screen is a text file named FKEY10.MNU, normally located in the C: \BATCH directory.

Rather than count characters as you repeatedly press the space bar, use the DEBUG editor's Fill command to make the prototype screen file. Program C-1 lists FKEY10.SCR, which creates the prototype screen display in Figure C-3 and places it within C: \BATCH \FKEY10.MNU. Be sure to include the single blank line after each of the lines beginning with *DB* near the end of the file. These blank lines terminate the Assemble commands begun two lines earlier.

Program C-1. The FKEY10.SCR file is a DEBUG input script that creates the prototype screen display file C: \BATCH \FKEY10.MNU for 10-option menus and a file for erasing a line, C: \BATCH \ BLANK77.INP.

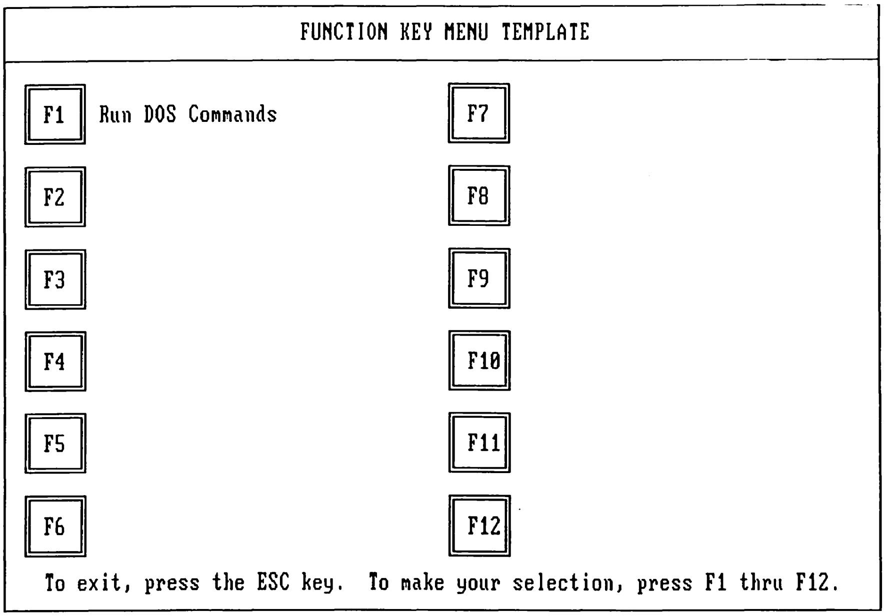
```
N C:\BATCH\FKEY10.MNU
R CX
0747
F 0100 L 0747 20
F 0100 L 0001 DA
F 0101 L 004D C4
F 014E L 0004 BF OD OA B3
F 019F L 0004 B3 OD OA B3
F 01FO L 0004 B3 OD OA B3
F 0241 L 0004 B3 OD OA C3
F 0245 L 004D C4
F 0292 L 0004 B4 OD OA B3
F 02E3 L 0004 B3 OD OA B3
F 0334 L 0004 B3 OD OA B3
F 0385 L 0004 B3 OD OA B3
F 03D6 L 0004 B3 OD OA B3
F 0427 L 0004 B3 OD OA B3
F 0478 L 0004 B3 OD OA B3
F 04C9 L 0004 B3 OD OA B3
F 051A L 0004 B3 OD OA B3
F 056B L 0004 B3 OD OA B3
F 05BC L 0004 B3 OD OA B3
F 060D L 0004 B3 OD OA B3
F 065E L 0004 B3 OD OA B3
F 06AF L 0004 B3 OD OA B3
F 0700 L 0004 B3 OD OA B3
F 0751 L 0004 B3 OD OA C3
```

```
F 0755 L 004D C4
F 07A2 L 0004 B4 0D 0A B3
F 07F3 L 0004 B3 0D 0A C0
F 07F7 L 004D C4
F 0844 L 0003 D9 0D 0A
F 0297 L 0006 C9 CD CD CD CD BB
F 02E8 L 0006 BA 20 46 31 20 BA
F 0339 L 0006 C8 CD CD CD CD BC
F 038A L 0006 C9 CD CD CD CD BB
F 03DB L 0006 BA 20 46 32 20 BA
F 042C L 0006 C8 CD CD CD CD BC
F 047D L 0006 C9 CD CD CD CD BB
F 04CE L 0006 BA 20 46 33 20 BA
F 051F L 0006 C8 CD CD CD CD BC
F 0570 L 0006 C9 CD CD CD CD BB
F 05C1 L 0006 BA 20 46 34 20 BA
F 0612 L 0006 C8 CD CD CD CD BC
F 0663 L 0006 C9 CD CD CD CD BB
F 06B4 L 0006 BA 20 46 35 20 BA
F 0705 L 0006 C8 CD CD CD CD BC
F 02BD L 0006 C9 CD CD CD CD BB
F 030E L 0006 BA 20 46 36 20 BA
F 035F L 0006 C8 CD CD CD CD BC
F 03B0 L 0006 C9 CD CD CD CD BB
F 0401 L 0006 BA 20 46 37 20 BA
F 0452 L 0006 C8 CD CD CD CD BC
F 04A3 L 0006 C9 CD CD CD CD BB
F 04F4 L 0006 BA 20 46 38 20 BA
F 0545 L 0006 C8 CD CD CD CD BC
F 0596 L 0006 C9 CD CD CD CD BB
F 05E7 L 0006 BA 20 46 39 20 BA
F 0638 L 0006 C8 CD CD CD CD BC
F 0689 L 0006 C9 CD CD CD CD BB
F 06DA L 0006 BA 20 46 31 30 BA
F 072B L 0006 C8 CD CD CD CD BC
A 02EF
DB 'Run DOS Commands'

A 07A9
DB 'To exit, press the Esc key. To make your selection, press F1
thru F10.'

A 01BD
DB 'FUNCTION KEY MENU TEMPLATE'

W
```

```
N C:\BATCH\BLANK77.INP
F 0100 L 004D 20
R CX
004D
W
Q
```

Another prototype screen format, FKEY12.MNU, is appropriate for systems with an Enhanced Keyboard. As you can see in Figure C-4, it offers 12 choices for the function keys 1–12. To accomodate the three additional lines of function key display, two blank lines were removed from the menu title area, as well as the separator line above the usage message.

Figure C-4. FKEY12.MNU contains the prototype screen display for twelve-option menus.



The FKEY12.SCR file in Program C-2 creates the 12-option prototype menu screen. Both FKEY10.SCR and FKEY12.SCR create an additional file, C: \BATCH \BLANK77.INP. Sending the BLANK77.INP file to the screen erases one line of text within a menu screen.

Program C-2. The FKEY12.SCR file is a DEBUG input script that creates the prototype screen display file C: \BATCH \ FKEY12.MNU for 12-option menus and a file for erasing a line, C: \BATCH \ BLANK77.INP.

```
N C:\BATCH\FKEY12.MNU
R CX
0747
F 0100 L 0747 20
F 0100 L 0001 DA
F 0101 L 004D C4
F 014E L 0004 BF OD OA B3
F 019F L 0004 B3 OD OA C3
F 01A3 L 004D C4
F 01F0 L 0004 B4 OD OA B3
F 0241 L 0004 B3 OD OA B3
F 0292 L 0004 B3 OD OA B3
F 02E3 L 0004 B3 OD OA B3
F 0334 L 0004 B3 OD OA B3
F 0385 L 0004 B3 OD OA B3
F 03D6 L 0004 B3 OD OA B3
F 0427 L 0004 B3 OD OA B3
F 0478 L 0004 B3 OD OA B3
F 04C9 L 0004 B3 OD OA B3
F 051A L 0004 B3 OD OA B3
F 056B L 0004 B3 OD OA B3
F 05BC L 0004 B3 OD OA B3
F 060D L 0004 B3 OD OA B3
F 065E L 0004 B3 OD OA B3
F 06AF L 0004 B3 OD OA B3
F 0700 L 0004 B3 OD OA B3
F 0751 L 0004 B3 OD OA B3
F 07A2 L 0004 B3 OD OA B3
F 07F3 L 0004 B3 OD OA C0
F 07F7 L 004D C4
F 0844 L 0003 D9 OD OA
F 01F5 L 0006 C9 CD CD CD CD BB
F 0246 L 0006 BA 20 46 31 20 BA
F 0297 L 0006 C8 CD CD CD CD BC
F 02E8 L 0006 C9 CD CD CD CD BB
F 0339 L 0006 BA 20 46 32 20 BA
F 038A L 0006 C8 CD CD CD CD BC
F 03DB L 0006 C9 CD CD CD CD BB
F 042C L 0006 BA 20 46 33 20 BA
```

```
F 047D L 0006 C8 CD CD CD CD BC
F 04CE L 0006 C9 CD CD CD CD BB
F 051F L 0006 BA 20 46 34 20 BA
F 0570 L 0006 C8 CD CD CD CD BC
F 05C1 L 0006 C9 CD CD CD CD BB
F 0612 L 0006 BA 20 46 35 20 BA
F 0663 L 0006 C8 CD CD CD CD BC
F 06B4 L 0006 C9 CD CD CD CD BB
F 0705 L 0006 BA 20 46 36 20 BA
F 0756 L 0006 C8 CD CD CD CD BC
F 021B L 0006 C9 CD CD CD CD BB
F 026C L 0006 BA 20 46 37 20 BA
F 02BD L 0006 C8 CD CD CD CD BC
F 030E L 0006 C9 CD CD CD CD BB
F 035F L 0006 BA 20 46 38 20 BA
F 03B0 L 0006 C8 CD CD CD CD BC
F 0401 L 0006 C9 CD CD CD CD BB
F 0452 L 0006 BA 20 46 39 20 BA
F 04A3 L 0006 C8 CD CD CD CD BC
F 04F4 L 0006 C9 CD CD CD CD BB
F 0545 L 0006 BA 20 46 31 30 BA
F 0596 L 0006 C8 CD CD CD CD BC
F 05E7 L 0006 C9 CD CD CD CD BB
F 0638 L 0006 BA 20 46 31 31 BA
F 0689 L 0006 C8 CD CD CD CD BC
F 06DA L 0006 C9 CD CD CD CD BB
F 072B L 0006 BA 20 46 31 32 BA
F 077C L 0006 C8 CD CD CD CD BC
A 024D
DB 'Run DOS Commands'

A 07A9
DB 'To exit, press the Esc key. To make your selection, press F1
thru F12.'

A 016C
DB 'FUNCTION KEY MENU TEMPLATE'

W
N C:\BATCH\BLANK77.INP
F 0100 L 004D 20

R CX
004D
W
Q
```

BLANK77.INP consists of 77 spaces (20H). Recall that the Batch Menu System screen lines are 79 characters wide. The first and last characters contain line-drawing characters from the IBM extended character set usually vertical bars (B3H). After positioning the cursor on the second character, a batch file can erase a line within a menu by copying BLANK77.INP to the screen, as illustrated by a portion of a Batch Menu System job stream:

```
:NOTYET
cursor 23 8
echo Sorry, that function is not active in this menu. Try another.
bell
cursor 23 1
getreply
copy c:\batch\blank77.inp/b con: > nul
goto :CHKKEY
```

Prototype menu job streams. The Batch Menu System is what its name implies. It is a screen menu system implemented via batch files. Using batch jobs for menu logic provides certain advantages:

- All the necessary software is part of MS-DOS, or within Appendix A and Appendix C of this book.
- It can be used on most MS-DOS machines because only standard MS-DOS facilites are employed.
- It is controlled by public environment variables, so it is easily customizaed.
- Anyone can understand and maintain it.

The Batch Menu System contains two prototype job streams. The first, MENU.BAT, defines various environment variables. FKEYMENU.BAT provides the framework for a menu job stream.

To simplify adapting the Batch Menu System to different disk directory structures, two environment variables control where the menus are located and where the batch files reside. To allow easy customization of individual preferences, such as screen color and cursor size, the menu system employs two additional environment variables. These are defined within the Batch Menu System entry point, the MENU.BAT file listed in Figure C-5.

Figure C-5. The MENU.BAT job stream defines environment variables to control the Batch Menu System. Then it runs the MAIN menu unless another was specified on the command line.

```
echo off
REM Batch Menu System initiator
set menu=%1
if "%menu%"=="" set menu=MAIN
if "%cursor%"=="" set cursor=cursor full
if "%colors%"=="" set colors=color 14
if "%batch%"=="" set batch=c:\batch\
if "%menudir%"=="" set menudir=c\batch\
:NEST
%cursor%
%colors%
command/c %menudir%%menu%:END
%cursor%
%colors%
exit
```

The normal way to start the Batch Menu System is to issue the command without parameters, as in **MENU**.

Referring to MENU.BAT in Figure C-5, you can see that the *%menu%* environment variable contains the value of the optional first command-line parameter. If no value is given, the environment variable is set to MAIN. This value is the name of the batch jobstream. The *%menudir%* environment variable specifies directory in which the Batch Menu System screens and jobstreams reside. The *%batch%* environment variable defines the location of the application and system jobstreams.

As MENU.BAT appears in Figure C-5, the MENU command would invoke C: \BATCH \MAIN.BAT. To run a different menu, merely specify it as a command line parameter. For instance, to start the menu system with C: \BATCH \DBASE.BAT, use the command MENU DBASE.

Figure C-6. The LOGON.BAT job stream provides a Batch Menu System *front-end* that provides different configurations for various users.

```
echo off
REM Batch Menu System initiator for multiple users
set user=%1
set menu=%2
REM allow quick-skip to default user
if "%user%"=="" goto :STANDARD
if "%user%"=="." goto :STANDARD
REM allow for reasonable capitalization variation
if "%user%"=="bob" goto :BOB
if "%user%"=="BOB" goto :BOB
if "%user%"=="Bob" goto :BOB
if "%user%"=="carol" goto :CAROL
if "%user%"=="CAROL" goto :CAROL
if "%user%"=="Carol" goto :CAROL
if "%user%"=="ted" goto :TED
if "%user%"=="TED" goto :TED
if "%user%"=="Ted" goto :TED
if "%user%"=="alice" goto :ALICE
if "%user%"=="ALICE" goto :ALICE
if "%user%"=="Alice" goto :ALICE
REM everyone else uses standard setup
goto :STANDARD
:BOB
if "%menu%"=="" set menu=FINANCE
if "%cursor%"=="" set cursor=cursor full
if "%colors%"=="" set colors=cls
if "%batch%"=="" set batch=c:\batch\
if "%menudir%"=="" set menudir=d:\finance\bob\
goto :NEST
:CAROL
if "%menu%"=="" set menu=DBASE
if "%cursor%"=="" set cursor=cursor small
if "%colors%"=="" set colors=color 2,0
if "%batch%"=="" set batch=c:\batch\
if "%menudir%"=="" set menudir=c:\batch\
```

```
goto :NEST
:TED
if "%menu%"=="" set menu=WORDSTAR
if "%cursor%"=="" set cursor=cursor full
if "%colors%"=="" set colors=color 14
if "%batch%"=="" set batch=c:\batch\
if "%menudir%"=="" set menudir=c:\batch\
goto :NEST
:ALICE
if "%menu%"=="" set menu=MAIN
if "%cursor%"=="" set cursor=cursor full
if "%colors%"=="" set colors=color 14,5
if "%batch%"=="" set batch=c:\batch\
if "%menudir%"=="" set menudir=c:\batch\
goto :NEST
:STANDARD
if "%menu%"=="" set menu=MAIN
if "%cursor%"=="" set cursor=cursor full
if "%colors%"=="" set colors=color 14
if "%batch%"=="" set batch=c:\batch\
if "%menudir%"=="" set menudir=c:\batch\
:NEST
%cursor%
%colors%
command/c %menudir%%menu%
:LOGOFF
set user=
set menu=
set batch=
set menudir=set cursor=cursor full
set colors=color 14
:END
%cursor%
%colors%
exit
```

Because the Batch Menu System begins with a *front-end* process, you can easily customize it for different users. A job stream could vary the environment values based upon who was running it. Perhaps one person uses monochrome *paper mode* (black letters on a white screen), while another prefers a block cursor and white letters on a black screen. Not only the directories, but also the initial menu jobstream may change. Each user can run with an appropriate list of menu options. Figure C-6 lists the LOGON.BAT job stream, which employs a command-line parameter to identify the operator.

The LOGON.BAT jobstream in Figure C-6 accepts two command-line parameters. The first becomes the *%user%* variable, which determines whether a custom or standard setup is required. Notice that both uppercase and lowercase versions of the names are checked. For each user name, the tests start with the most likely format (all lowercase in this instance) and proceed to the least likely.

The most efficient sequence to check for various users is to put the most frequent user first in the list. Many installations prefer the simple approach, listing them alphabetically. To speed user-name checking, both a dot (.) and no name use the :STANDARD configuration. The dot serves as a place-marker. It allows a :STANDARD user to specify a different menu in the second command line parameter.

The second parameter loads the %menu% environment variable. It functions exactly as the first parameter for the MENU jobstream in Figure C-5. For instance, a :STANDARD user may elect to run the C: \BATCH \MAINT.BAT menu with this command:

LOGON . MAINT

Normally, BOB runs D: \FINANCE \BOB \FINANCE.BAT, but he could elect to run another of his menus. To run D: \FINANCE \BOB \RECAP.BAT, he would enter **LOGON BOB RECAP**.

Rather than requiring a command-line user-name, the LOGON.BAT job stream could employ its own menu and screen display. The operator would then press the appropriate function key to define the %user% environment variable. To retain the flexibility of the command-line approach, the log-on menu could be invoked only within the :STANDARD processing of the LOGON

jobstream. Then users could still do a quick log-on.

You may wish to replace the :END commands in Figures C-7 and C-8 with an invocation of the ROOT.BAT job stream. ROOT is explained in the System Support Job Streams portion of this appendix. The code would then be

```
TECH
:END
root
end TECH
```

Each menu jobstream is built from the prototype FKEYMENU.BAT file, listed in Figure C-7. This job stream contains all the commands necessary to run a single-option menu. Both the prototype menu screen displays, in Figures C-3 and C-4, contain a predefined menu option F1, *Run DOS Commands*. The :F1 label within the FKEYMENU jobstream implements this function.

Figure C-7. The FKEYMENU.BAT job stream is the prototype Batch Menu job stream.

```
echo off
REM FKEYMENU is the prototype Batch Menu job stream
set menu=FKEY10
:MENU
set task=
cursor no
%colors%
copy %menudir%%menu%.mnu/b con: > nul
:GETKEY
getreply
:CHKKEY
if errorlevel 27 if not errorlevel 28 goto :END
if errorlevel 187 if not errorlevel 188 goto :F1
if errorlevel 188 if not errorlevel 189 goto :F2
if errorlevel 189 if not errorlevel 190 goto :F3
if errorlevel 190 if not errorlevel 191 goto :F4
if errorlevel 191 if not errorlevel 192 goto :F5
if errorlevel 192 if not errorlevel 193 goto :F6
if errorlevel 193 if not errorlevel 194 goto :F7
if errorlevel 194 if not errorlevel 195 goto :F8
if errorlevel 195 if not errorlevel 196 goto :F9
if errorlevel 196 if not errorlevel 197 goto :F10
if errorlevel 133 if not errorlevel 134 goto :F11
```

```
if errorlevel 134 if not errorlevel 135 goto :F12
bell
goto :GETKEY
:F1
set task=command
goto :NEST
:F2
:F3
:F4
:F5
:F6
:F7
:F8
:F9
:F10
:F11
:F12
:NOTYETcursor 23 8
echo Sorry, that function is not active in this menu. Try another.
bell
cursor 23 1
getreply
copy %menudir%blank77.inp/b con: > nul
goto :CHKKEY
:NEST
%cursor%
%colors%
command/c %task%
goto :MENU
:END
%cursor%
%colors%
exit
```

The labels for the unimplemented options stack up over the :NOTYET procedure. It displays a message below the usage message and generates an audible tone. :NOTYET then repositions the cursor (indented from the left margin by one column) prepatory to erasing the message later. Since the CURSOR program sets the ERRORLEVEL, it must run before GETREPLY waits for the keypress. Once the operator presses a key, the batch file erases the message line. It proceeds to check for a valid keypress by jumping to the label :CHKKEY.

If the Esc key is pressed to select a menu option, the job stream jumps to its :END process. It then clears the screen and exits, returning control to its parent process, the one that invoked it. Parent processes are discussed in the next section.

The FKEYMENU.BAT job stream provides the structure for a viable Batch Menu System. It uses environment variables for adaptability and efficiency. It provides a polished, professional feel. But it is just a prototype, the starting point for customization.

Creating custom menus. You can easily create your own custom menus from the prototype menu screen displays, FKEY10.MNU and FKEY12.MNU, plus the prototype menu job stream, FKEYMENU.BAT. Customizing menus entails five steps:

- Design: Select the menu options that will appear upon each menu. Consider how each option will behave, and what subordinate processes it will involve. Consider grouping related options into a menu, such as disk formatting, disk duplication, and file copying.
- Prototype: Create a small batch file that is a stand-alone version of each menu option. Run it, test it, and tweak it.
- Display: Create or update the menu screen displays. Use an editor program or the DEBUG input scripts in Programs C-3 and C-4 to modify a copy of one of the prototype menu display screens.
- Implement: Insert the setup and invocation commands into the appropriate portion of a copy of the prototype menu job stream.
- Enjoy: Run the menu and select the menu option; experience the thrill of creating something that works.

The Application Job Streams listed in the next section all began as prototype job streams. Some of them are very linear, providing no choices about how their task is run. Others use batch file interaction to vary their behavior. We recommend that all Batch menu options themselves be batch files. This keeps the menus smaller and faster. It also allows you to invoke a common job stream from multiple menus.

Within a menu screen display, two items vary. The menu title, at the top of the screen, should not exceed 65 characters in width to avoid a cluttered look. This length is also the maximum allowable DB statement supported by the DEBUG Assemble command. The individual menu option titles are limited by the adjacent line-drawing characters. Each menu option Title may not exceed 28

characters. Program C-3 provides a template for inserting the menu title and menu option titles into ten-option menu screen displays. It creates TEST.MNU in the C: \BATCH directory. If your menu directory ("%menudir%") is different, change the script accordingly.

Program C-3. The DEBUG input script FKEY10TX.SCR modifies a copy of the FKEY10.MNU ten-option prototype menu display by adding titles.

```
N   C:\BATCH\FKEY10.MNU
L
N   C:\BATCH\TEST.MNU
A   01A9
;   TITLE 12345678901234567890123456789012345678901234567890123456789012345
DB '                                                                  '
A   02EF
;   F1       1234567890123456789012345678
DB           'Run DOS Commands            '
A   03E2
;   F2       1234567890123456789012345678
DB           '                            '
A   04D5
;   F3       1234567890123456789012345678
DB           '                            '
A   05C8
;   F4       1234567890123456789012345678
DB           '                            '
A   06BB
;   F5       1234567890123456789012345678
DB           '                            '
A   0315
;   F6       1234567890123456789012345678
DB           '                            '
A   0408
;   F7       1234567890123456789012345678
DB           '                            '
A   04FB
;   F8       1234567890123456789012345678
DB           '                            '
A   05EE
;   F9       1234567890123456789012345678
DB           '                            '

A   06E1
;   F10      1234567890123456789012345678
DB           '                            '
W
Q
```

The FKEY10TX.SCR script in Program C-3 loads the prototype menu screen into memory. It then names the file TEST.MNU, which it will subsequently write into the directory, C: \BATCH. A series of Assemble commands inserts the menu title and the various menu option descriptions. Between each Assemble command

and its ensuing DB (Define Byte) statement, comments appear. After each DB line, a blank line ends the Assemble command.

DEBUG's built-in assembly language translator treats all text after the first semicolon (;) in the line as a comment. The comments within the file both identify each field (TITLE, F9) and provide a ruler to insure that the entire field area is defined. You will probably want to center your menu title. The ruler makes this chore easier.

After all the text is assembled, the script writes out the new menu screen. Program C-4 contains the FKEY12TX.SCR script, which performs the same functions as FKEY10TX.SCR, but for twelve-option menus.

Program C-4. The DEBUG input script FKEY12TX.SCR modifies a copy of the FKEY12.MNU twelve-option prototype menu display by adding titles.

```
N   C:\BATCH\FKEY12.MNU
L
N   C:\BATCH\TEST.MNU
A   0158
;    TITLE 12345678901234567890123456789012345678901234567890123456789012345
DB  '                                                                       '
A   024D
;   F1      1234567890123456789012345678
DB          'Run DOS Commands            '
A   0340
;   F2      1234567890123456789012345678
DB          '                            '
A   0433
;   F3      1234567890123456789012345678
DB          '                            '
A   0526
;   F4      1234567890123456789012345678
DB          '                            '
A   0619
;   F5      1234567890123456789012345678
DB          '                            '
A   070C
;   F6      1234567890123456789012345678
DB          '                            '
A   0273
;   F7      1234567890123456789012345678
DB          '                            '
A   0366
;   F8      1234567890123456789012345678
DB          '                            '
A   0459
;   F9      1234567890123456789012345678
DB          '                            '
A   054C
```

```
;   F10     1234567890123456789012345678
DB          '                             '
A   063F
;   F11     1234567890123456789012345678
DB          '                             '
A   0732
;   F12     1234567890123456789012345678
DB          '                             '

W
Q
```

Once the screen has been defined, see how it looks. To view your handiwork on the screen, enter **TYPE C: \BATCH \ TEST.MNU**.

You'll want to make the corresponding menu job stream. Copy the FKEYMENU.BAT to create a TEST.BAT starting-point in the menu directory:

COPY C: \BATCH \FKEYMENU.BAT C: \BATCH \TEST.BAT

Menu options invoke either *child* or *sibling* tasks. When the Esc key is pressed to select the menu option, the menu job stream jumps to the :END process, where it clears the screen and returns to its parent process.

In the prototype job streams, either MENU.BAT or LOGON.BAT is the parent of MAIN.BAT. When MAIN runs a child process, it regains control at the end of the child task. When MAIN runs a sibling process, either MENU or LOGON get control when the sibling task completes.

For each active menu option, the environment variable *%task%* is defined as the command line to be run. The :NEST procedure will invoke a secondary command processor, telling it the %task% to perform. This is a child process. It returns control to the menu job stream when it finishes. The menu then jumps to the :MENU label, where it redraws the display and awaits a selection. For instance, to have menu option F2 run the SUPPORT menu as a child process and menu option F3 use the *1-2-3* batch job stream as a child, change the menu job stream to:

:F2
SET TASK=%MENUDIR%SUPPORT
GOTO :NEST
:F3
SET TASK=%BATCH%1-2-3
GOTO :NEST
:F4

Notice that menus are in the %menudir% menu directory and application job streams are in the %batch% directory. These two environment variables may point to a single directory. If they do, maintain the distinction between the menu and batch directory variables to allow private menu directories without rewriting everything.

If a procedure is not nested, a slightly different invocation is used. For instance, suppose that menu option :F2 invoked a sibling menu, rather than a child menu. Use :NEST to run a child process; directly invoke a sibling menu. To cause :F2 to call the SUPPORT menu as a sibling, code:

```
:F2
SET TASK=%MENUDIR%SUPPORT
%TASK%
:F3
```

This code fragment could avoid using the %task% environment variable altogether. If your subordinate task has no need to query the %task% environment variable to alter its behavior, you may elect to write:

```
:F2
%MENUDIR%SUPPORT
:F3
```

Edit TEST.BAT to include child or sibling invocations of the prototype job streams you previously tested from the command line. Or use the ready-to-run job streams in the following sections of this appendix. When you are ready to test your menu, enter either LOGON.TEST or MENU TEST.

When you have a working menu, copy it to its final directory and filename. Then make it a child or sibling process within an existing menu or two. It will then be integrated into your Batch Menu System.

Application Job Streams

The following batch files invoke various commercial application software. They have been tested to run unaltered from the Batch Menu System, GEM, or the MS-DOS command line. To use them with *Windows* or *TopView*, copy the target application's Program Information File (PIF) and give it the name of the batch file.

For instance, the 1-2-3.BAT job stream runs Lotus Development's *1-2-3* package. To enable using 1-2-3.BAT from within *Windows,* enter the commands

```
C:
CD C:\WINDOWS\PIF
COPY LOTUS.PIF 1-2-3.PIF
CD ..
WIN
```

You must alter a few entries within the new file (1-2-3.PIF). Use the PIFEDIT.EXE program to make the following changes:

- Program name: The batch file name, such as 1-2-3.BAT.
- Program parameters: Clear it, as the job streams take no parameters.
- Initial directory: The batch file directory, such as C:\WINDOWS.
- Close window on exit: Activate it (make it *Yes*).

The job streams within this section are:

- ARCHIVE.BAT creates SoftStrip paper archives of files on a LaserJet printer.
- BACKUP.BAT performs either an incremental or total backup via *COREFast.*
- CHECKDSK.BAT verifies disk drive integrity via MS-DOS CHKDSK and *Norton*'s DiskTest.
- MAKEDISK.BAT formats disks as either high density in drive A or low density in drive B.
- RESTORE.BAT performs *COREFast* restore.
- SUPRCALC.BAT runs *SuperCalc3* with a mouse menu, plus *SideWays,* both for a dot-matrix printer.
- TIMELINE.BAT runs *TimeLine* with a mouse menu.
- VENTURA.BAT runs *Ventura Publisher 1.1* for Hercules.
- WORDSTAR.BAT invokes unadorned *WordStar 2000.*

These job streams assume that the applications reside within the directories recommended by the various software publishers. Although not required, some of the batch files assume that the C:\HDWR\POP-UPS directory exists. If it is present, this directory would contain a mouse menu system. Logitec, Microsoft, and Mouse Systems all provide mouse menu software to "mouse-ify"

programs. Mouse menus translate mouse movement into cursor-arrow keypresses and mouse-button presses into keystroke sequences.

These job streams assume that a Mouse Systems' PC Mouse pointing device is installed upon serial port COM2:. This assumption appears in job streams as:

- MOUSESYS /2 activates mouse menus on COM2.
- MOUSESYS /D disables mouse menus.
- M_programname loads mouse menu for program name (M_LOTUS is the mouse menu for LOTUS).

If your mouse hardware is different, alter references to MOUSESYS and all programs beginning with M_, such as M_LOTUS. If your mouse software is located within another directory, change all occurrences of C: \HDWR \POP-UPS. If you have no mouse installed, either consider getting one, or remove the mouse directory and mouse software commands from the batch files.

ARCHIVE.BAT: Offline backup on SoftStrips. Cauzin Systems manufactures an interesting device, the SoftStrip Reader. This lightweight 16 × 4 × 4–inch device communicates via your computer serial port. It translates printed strips into binary computer data. The strips contain data encoded in the proprietary SoftStrip format.

SoftStrips may be printed by a dot-matrix or laser printer under the control of Cauzin software. A single 8½ × 11 inch page of SoftStrips can hold about 8K of data. Dot-matrix printing is too slow to be useful, but the Hewlett-Packard LaserJet provides fast paper backups with Cauzin's *Archivist* software at the helm.

The ARCHIVE.BAT job stream in Figure C-8 runs the *Archivist* program to print the entire contents of the scratch-pad directory (C: \SCRATCH) on the LaserJet as machine-readable SoftStrips. The job stream uses the *Xtree* directory manager program to move the files into the scratch-pad area. After printing the files, the job stream removes the archived files from C: \SCRATCH.

Figure C-8. ARCHIVE.BAT prints all the files in the C: \SCRATCH scratch-pad directory on SoftStrips.

```
echo off
REM uses SoftStrips to archive files offline
cls
```

```
echo This procedure puts files that you select into archival (long-term)
   storage.
echo It first lets you copy the files into a scratch-pad area on the hard
   disk.
echo Next, the files are printed on the laser printer in a machine-
   readable form.
echo Finally, the scratch-pad area is erased.
echo+
echo Do you want to do this now?
set adry=:PREPLASER
set adrn=:END
set adre=:END
goto :YESORNO
:PREPLASER
echo Verify that the laser printer is connected...
pause
mode com1:9600,n,8,1,p
cls
echo Are the files to be archived already in the directory
   C:\SCRATCH?
set adry=:PRINTEM
set adrn=:GETEM
goto :YESORNO
:YESORNO
echo Indicate YES or NO by pressing the Y or N key.
echo Press the ESC key to quit.
:QUERYgetreply.com
if errorlevel 27 if not errorlevel 28 goto %adre%
if errorlevel 78 if not errorlevel 79 goto %adrn%
if errorlevel 89 if not errorlevel 90 goto %adry%
bell.com
goto :QUERY
:GETEM
echo Copy the files you wish to archive into the directory
   C:\SCRATCH. Do this
echo by first "tagging" and then copying them. You may wish to
   delete
echo them once they are copied.
echo+
pause
echo Y | del c:\scratch\*.*
xtree
:PRINTEM
cls
```

```
set CAUZIN=/P:HP,COM1 /L
c:\cauzin\la c:\scratch\*.*
cls
echo Did you successfully create the laser archive?
set adry=:SCRUB
set adrn=:END
goto :YESORNO
:SCRUB
echo Y | del c:\scratch\*.*
:END
exit
```

If you include the ARCHIVE job stream in a batch menu, a possible menu title for the DEBUG input script is:

```
; FN   1234567890123456789012345678
DB   'CREATE SOFTSTRIP ARCHIVE '
```

BACKUP.BAT: System-wide backup. While every application should include its own backup plan, system-wide backups are also part of defensive computing. The BACKUP job stream in Figure C-9 uses Core International's *COREFast* backup utility. The job stream queries the operator whether a complete or incremental backup is desired. The operator's response is translated into the presence or absence of the */M* command-line parameter for the backup program.

Figure C-9. BACKUP.BAT queries whether to perform an incremental or total system backup.

```
echo off
rem perform complete or incremental backup via COREFast
cls
echo You may choose to back up either the entire contents of the hard
   disk,
echo or only those files that have been altered since the last backup.
   You
echo should perform a complete backup periodically (at least once a
   month).
echo The partial backup is an incremental backup. It is a lot faster
   than a
echo complete backup, but you need to retain all of your incremental
   backups
echo since your last complete backup. When you make a complete
   backup, the
```

```
echo old incremental backups become obsolete.
echo+
set backup=
echo Do you want to perform an complete backup?
echo+
:YESORNO
echo Indicate YES or NO by pressing the Y or N key.
echo Press the ESC key to quit.
:QUERY
getreply.com
if errorlevel 27 if not errorlevel 28 goto :END
if errorlevel 78 if not errorlevel 79 goto :INCREMENTAL
if errorlevel 89 if not errorlevel 90 goto :COMPLETE
bell.com
goto :QUERY
:INCREMENTAL
set backup=/m
:COMPLETE
REM all subdirectories, high-speed, read-after-write check
cfback c:\*.* a: /s /r %backup%
set backup=
:END
exit
```

If you include the BACKUP job stream in a batch menu, a possible menu title for the DEBUG input script is

```
; FN   1234567890123456789012345678
DB   'SYSTEM BACKUP               '
```

CHECKDSK.BAT: Check hard disk integrity. A good habit to acquire is performing a periodic check on the health of your hard disk. The CHECKDSK batch file in Figure C-10 scans the File Allocation Table (FAT), gathering any lost clusters of file space into temporary files, which it then deletes. It invokes the DT (DiskTest) program from the *Norton Utilites* package to detect and map any bad sectors.

Figure C-10. CHECKDSK.BAT performs an integrity test of the hard disk.

```
echo off
REM recover lost files and mark bad sectors
chkdsk c: /f
del c:\file????.chk
dt c: /d
```

If you include the CHECKDSK job stream in a batch menu, a possible menu title for the DEBUG input script is

```
; FN   1234567890123456789012345678
DB   'HARD DISK CHECKUP
```

MAKEDISK.BAT: Formatting floppy disks. Many AT-style computers are equipped with two disk drives. While drive A is almost always a high-capacity 1.2Mb device, the second drive is often for low-density 360K disks.

Unless you use a Panasonic JU4752-AEG drive, low-density floppy disks formatted on a high-density drive are often unreadable on a PC or XT system. The MAKEDISK job stream in Figure C-11 asks the operator whether it should create high- or low-density disks. It then uses the appropriate drive for the desired density.

Figure C-11. MAKEDISK.BAT lets the operator select whether to format high- or low-density disks.

```
echo off
REM format floppy disks; A: is high-density, B: is low
cls
echo This jobsteam prepares new or used disks for use.
echo You may make disks either low density or high density.
echo+
echo High density disks can hold 1200K of data, but they
echo are only usable by AT-type computers.
echo+
echo If you want to be able to use the disk on computers
echo other than AT-type (PC or XT), then you must prepare
echo Low density 360K disks for use. These also work
echo on AT-type computers.
echo+
ask Do you want to create a high density disk?
:YESORNO
echo+
echo Indicate YES or NO by pressing the Y or N key.
echo Press the ESC key to quit.
:QUERY
getreply.com
if errorlevel 27 if not errorlevel 28 goto :END
if errorlevel 78 if not errorlevel 79 goto :LOW
if errorlevel 89 if not errorlevel 90 goto :HIGH
bell.com
```

```
goto :QUERY
:LOW
format b: /4
goto :END
:HIGH
format a:
:END
exit
```

If you include the MAKEDISK job stream in a batch menu, a possible menu title for the DEBUG input script is

```
; FN  1234567890123456789012345678
DB   'INITIALIZE DISKS           '
```

RESTORE.BAT: Restore from backup. Core International's *COREFast* program provides both command-line and menu-driven modes. The RESTORE job stream in Figure C-12 takes advantage of this.

If the operator indicates that a complete restore is necessary, the job stream uses a command-line to do it because no file selection is required. Conversely, if the operator wants to perform a "pick-and-choose" partial restore, the job stream invokes the interactive version of *COREFast*.

Figure C-12. RESTORE.BAT reads a *COREFast* backup set to restore files.

```
echo off
rem restore from COREFast backup
cls
echo You may choose to restore either the entire contents of the
   backup set,
echo or only those subdirectories that you indicate. You would
   perform a
echo complete restore only when you are trying to recover from a hard
   disk
echo failure. You would perform a partial restore when you wish to
   undo
echo the changes to files associated with a single task, such as
   spreadsheets.
echo+
echo Do you want to perform a complete restore?
echo+
:YESORNO
echo Indicate YES or NO by pressing the Y or N key.
```

```
echo Press the ESC key to quit.
:QUERY
getreply.com
if errorlevel 27 if not errorlevel 28 goto :END
if errorlevel 78 if not errorlevel 79 goto :PARTIAL
if errorlevel 89 if not errorlevel 90 goto :COMPLETE
bell.com
goto :QUERY
:PARTIAL
REM go interactive
corefast
goto :END
:COMPLETE
REM complete file-by-file high-speed, overwrites read-only files
cfrest a: \*.* c: /s /r
:END
exit
```

If you include the RESTORE job stream in a Batch Menu, a possible menu title for the DEBUG input script is

```
; FN    1234567890123456789012345678
DB    'RESTORE FROM BACKUP    '
```

SUPRCALC.BAT: *SuperCalc3* and *SideWays*. Software Associate's *SuperCalc* is a spreadsheet program with origins in the eight-bit CP/M world. The package has kept pace with MS-DOS technology. It provides very good graphics and a variety of data interchange formats. Mouse Systems includes a mouse menu for *SuperCalc*.

The SUPRCALC job stream in Figure C-13 first selects the parallel dot-matrix printer as the default device. Then it logs into the mouse directory. It activates the mouse menu interpreter and loads the *SuperCalc* menu. After logging into the *SuperCalc* program directory, it runs the spreadsheet utility. When *SuperCalc* exits, the job stream disables the mouse menu interpreter, activates the sideways-printing program, and reselects the serial printer as the default device.

Figure C-13. SUPRCALC.BAT uses the dot-matrix printer and mouse menus. The job stream allows sideways printing of spreadsheets.

```
echo off
REM SuperCalc uses parallel dot-matrix printer
mode lpt1:
c:
cd c:\hdwr\pop-ups
mousesys /2
m_sc3
c:
cd c:\sc3
sc3
mousesys /s
sideways
REM default back to serial laser printer (COM1: with DTR handshake)
mode com1:9600,n,8,1,p
mode lpt1:=com1
exit
```

If you include the SUPRCALC job stream in a batch menu, a possible menu title for the DEBUG input script is

```
; FN   1234567890123456789012345678
DB   'SPREADSHEETS                '
```

TIMELINE.BAT: Project management. Breakthough Software's *TimeLine* is a project management package. It provides for extensive scheduling and analysis. Mouse Systems includes a mouse menu for *TimeLine*. The TIMELINE job stream in Figure C-14 first logs into the mouse directory. It activates the mouse menu interpreter and loads the *TimeLine menu*. It then logs into the *TimeLine* program directory. After warning the operator about the noticeable delay of loading *TimeLine*, the batch file activates the program. When *TimeLine* exits, the job stream deactivates the mouse menu interpreter.

Figure C-14. TIMELINE.BAT grafts a mouse menu onto the *TimeLine* package.

```
echo off
REM TimeLine
cls
c:
```

```
cd c:\hdwr\pop-ups
mousesys /2
m_timeln
c:
cd c:\tl
echo One moment, reading TIMELINE.EXE . . .
cls
timeline
mousesys /s
```

If you include the TIMELINE job stream in a batch menu, a possible menu title for the DEBUG input script is

```
; FN  1234567890123456789012345678
DB   'PROJECT MANAGEMENT  '
```

VENTURA.BAT: Desktop publishing. *Xerox Desktop Publisher: Ventura Edition* is a full-featured publishing package for MS-DOS computers. *Version 1.1 Ventura* offers full typographic control and built-in mouse support. The VENTURA job stream in Figure C-15 invokes *Ventura 1.1* for use with a Hercules Graphics Card.

Figure C-15. VENTURA.BAT runs *Ventura Publisher* version 1.1 using a Hercules Graphics Card.

```
echo off
REM Ventura Publisher 1.1 with Hercules
c:
cd c:\ventura
drvrmrgr vp %1 /s=sd_herc5.ega\m=01
exit
```

If you include the VENTURA job stream in a batch menu, a possible menu title for the DEBUG input script is

```
; FN  1234567890123456789012345678
DB   'DESKTOP PUBLISHING      '
```

WORDSTAR.BAT: Wordprocessing. MicroPro's *WordStar* is a wordprocessing standard for microcomputers. Their *WordStar 2000* is a top-of-the-line product. It supports a variety of printers and offers document-annotation features.

The WORDSTAR job stream in Figure C-16 logs into a *WordStar 2000* subdirectory, WS2000\LETTERS. It then relies upon the standard program search PATH including the parent directory

(..) to find the WS2.EXE program. When the operator exits *WordStar*, the job stream removes any backup (.BAK) files from the then-current directory, freeing the operators of that responsibility.

Figure C-16. WORDSTAR.BAT runs an unadorned *WordStar 2000* from a subdirectory, relying upon the PATH to find the program overlays.

```
echo off
REM WordStar 2000
c:
cd c:\ws2000\letters
ws2
del *.bak
exit
```

If you include the WORDSTAR job stream in a batch menu, a possible menu title for the DEBUG input script is

```
; FN   1234567890123456789012345678
DB    'WORDPROCESSING              '
```

System Support Job Streams

The following batch files are invoked from the MS-DOS command line. They do not require the Batch Menu system, although they may be called from within it via the *Run DOS Commands* option.

The job streams within this section are:

- ASM.BAT, which assembles and links programs.
- CLR.BAT, which resets the screen to standard colors and cursor type.
- DIRS.BAT, which generates a sorted subdirectory list.
- DTG.BAT, which determines the system time stamp.
- ROOT.BAT, which logs all fixed drives into their root directories.
- SDIR.BAT, which generates a sorted file directory list.
- SET-CWD.BAT, which defines the current working directory.

ASM.BAT: Assemble and link. The Microsoft Macro Assembler, MASM.EXE, and its companion programs, EXE2BIN.EXE and LINK.EXE, indicate their success or failure by setting the system ERRORLEVEL. This section presents three job streams which use ERRORLEVEL when running MASM:

- ASM.BAT creates .OBJ object and .LST listing files.
- ASM2COM.BAT creates .COM programs, no listing.
- ASM2EXE.BAT creates .EXE programs, no listing.

These three job streams, listed below, take as their single parameter the filename of the Assembler source code. You may include a drive and path specifier, but do not include any filename extension:

- ASM COLOR
- ASM2COM A:CURSOR
- ASM2EXE C: \MASM \VIDTYPE

ASM.BAT runs MASM to create an .OBJ object file. It generates a .LST listing file including the symbol map. Enter it as shown:

```
MASM %1,%1,%1;
```

ASM2COM assembles and links to create a .COM program. It removes all interim files:

```
MASM %1,%1,NUL.LST -MX;
IF NOT ERRORLEVEL 1 LINK %1,%1;
IF EXIST %1.OBJ DEL %1.OBJ
IF NOT ERRORLEVEL 1 EXE2BIN %1
IF EXIST %1.EXE DEL %1.EXE
IF NOT ERRORLEVEL 1 DEL %1.COM
IF NOT EXIST %1.COM REN %1.BIN %1.COM
```

ASM2EXE assembles and links to create an .EXE program.

```
MASM %1,%1,NUL.LST -MX;
IF NOT ERRORLEVEL 1 LINK %1,%1;
IF EXIST %1.OBJ DEL %1.OBJ
```

CLR.BAT: Clear the screen. Although another version has appeared elsewhere within this book, the CLR job stream here is compatible with the Batch Menu System. CLR.BAT, listed in Figure C-17, uses existing environment variables to define its colors and cursor size. If the variables do not exist, it defines them.

Figure C-17. This version of the CLR.BAT job stream uses the same environment variables employed by the Batch Menu System.

```
echo off
rem insure standard screen is defined, then clear it
if "%cursor%"=="" set cursor=cursor full
if "%colors%"=="" set colors=color 14,5
%cursor%
%colors%
```

DIRS.BAT: Directories sort. This short job stream creates a sorted listing of the subdirectories on the specified disk drive directory. DIRS excludes all data file references from the list during the sort step. It takes zero, one, or two parameters:

- Zero parameters (DIRS) displays a paginated, sorted list of the subdirectories of the current directory (.).
- One parameter (DIRS A:) displays a paginated, sorted list of the subdirectories of the matching area (A:).
- Two parameters (DIRS \UTIL UTILDIR.LST) writes a sorted list of the subdirectories of the matching area (\UTIL \) to a device or file (UTILDIR.LST).

Full drive pathing is supported for the first parameter, the *matching area* specification. Either filenames or output devices are supported for the *destination file* specification, the second parameter. You must enter a matching area specification to indicate a destination file, but it can be the dot (.)—or current—directory:

```
DIRS . THISAREA.DIR
```

Figure C-18 lists the batch file. Notice that the *IF* test uses a GOTO. This is because a conditional statement cannot pipe.

Figure C-18. The DIRS.BAT job stream puts a sorted subdirectory list either on the screen or into a file.

```
echo off
REM sorted subdirectory
REM following line uses goto because conditionals do not pipe
if "%1"=="" goto :SCREEN
dir %1 | find "<" | sort > %2
goto :END
:SCREEN
dir %1 | find "<" | sort | more
:END
exit
```

DTG.BAT: Date-time group. The DTG batch file reads and displays the date and time from the system clock. Aside from turning your computer into an unwieldy wristwatch, this job stream can be merged into other batch files to create a *timestamp*.

The DTG job stream has one optional parameter—its destination file. To display the date and time upon the screen, enter the job stream name with no parameters:

DTG

Entering a destination filename parameter saves the timestamp to a file. If you wish to determine when different portions of a lengthy job stream began, you can create a series of timestamp files. At pertinent points within the job stream, you can insert commands to *nest* the DTG procedure, such as:

COMMAND/C DTG C:\SCRATCH\STEP4.DTG

Two interesting techniques are evident within this short batch file. Note the double output-redirection symbol (>>) in the fourth line of Figure C-19. It appends the time line to the end of the output file created by the single symbol (>) in the preceding date line.

Figure C-19. The DTG.BAT job stream determines the current date and time.

```
echo off
REM determine date-time group timestamp
if "%1"=="" goto :SCREEN
echo+ | more | date | find "C" > %1
echo+ | more | time | find "C" >> %1
goto :END
:SCREEN
echo+ | more | date | find "C"
echo+ | more | time | find "C"
:END
exit
```

The ECHO+ command generates as its output a space, an Enter, and a line feed. The initial space causes an input error for both the DATE and TIME programs. The MORE filter, however, sends a single Enter before passing on its input stream. Since MORE needs something as input, ECHO supplies it. The DATE and TIME filters only look at the initial Enter from MORE. They exit before reading

the ensuing ECHO data stream.

ROOT.BAT: Log into root directories. In a multidrive system, it's easy to lose track of which is the current directory on all of the hard disk drives. When the default drive is not logged into the expected directory, file problems that can occur on that other drive are:

- Files copied to the wrong directory
- Files not found
- Incorrect relative paths

The ROOT job stream listed in Figure C-20 alleviates the problem. It logs every fixed-media disk drive (no Drive Not Ready errors) into its root directory. The batch file begins with the last drive first, so it ends up logged into the root directory of the first fixed drive.

Figure C-20. ROOT.BAT logs into the root directory of all hard disk drives at once and then clears the screen.

```
echo off
REM move to root (level 0) directory on all hard drives
REM e:
REM cd e: \
REM d:
REM cd d: \
c:
cd c: \
clr
```

The last line of the ROOT.BAT file in Figure C-20 invokes another batch file, CLR.BAT in Figure C-17. If you prefer, you may elect either to replace the CLR invocation with a CLS command to clear the screen, or to delete this last line entirely.

As listed, the ROOT job stream only logs into hard disk drive C:. The command lines for drives D: and E: are merely batch-file comments (REM statements). The comments serve to remind you to log from last-to-first. To activate additional drives, merely edit out the REM preamble of the appropriate command lines.

Even on a single-drive system, the ROOT command is a handy way to get to a standard starting or stopping point. If you step away from your machine, you may not want to leave your directories naked and exposed. The inadvertent errors of someone using your system to erase everything from their floppy disk with the

command DEL *.* would not hurt a root directory laden only with subdirectories and read-only copies of COMMAND.COM, AUTOEXEC.BAT, and CONFIG.SYS. Can the same be said about your data directories?

Practice defensive computing. End all your batch job streams with a ROOT command. Invoke the separate batch procedure, rather than duplicating its code within many batch files. Then you won't have to change your job streams when you add more fixed-media drives or directory passwords.

SDIR.BAT: Sorted data directory. This short job stream creates a sorted directory listing of the specified disk drive directory. SDIR excludes all subdirectory references from the list during the sort step. It takes zero, one, or two parameters:

- Zero parameters (SDIR) displays a paginated, sorted list of the current directory (*.*).
- One parameter (SDIR A:*.BAT) displays a paginated, sorted list of the matching files (*.BAT).
- Two parameters (SDIR \UTIL UTILDIR.LST) writes a sorted list of the matching files (\UTIL *.*) to a device or file (UTILDIR.LST).

Full, ambiguous filenames are supported for the first parameter, the *matching files* specification. Either filenames or output devices are supported for the *destination file* specification, the second parameter. Full pathing is supported for both parameters. You must enter a matching files specification to indicate a destination file, but it can be the dot (.)—or current—directory:

SDIR . THISAREA.DIR

Figure C-21 lists the batch file. Notice that the IF test uses a GOTO. This is because a conditional statement cannot pipe.

Figure C-21. The SDIR.BAT job stream puts a sorted data directory list either on the screen or into a file.

```
echo off
REM sorted data directory
REM following line uses goto because conditionals do not pipe
if "%2"=="" goto :SCREEN
dir %1 | find/v "<" | sort > %2
goto :END
:SCREEN
```

```
dir %1 | find/v "<" | sort | more
:END
exit
```

SET-CWD.BAT: Set current working directory. Use this job stream to define the current drive and directory as your current working directory. Run it with the command:

SET-CWD

It creates C: \BATCH \CWD.BAT, which contains the commands to log into the current disk drive and directory. For instance, if you ran SET-CWD.BAT while logged into directory D: \WS4 \LETTERS, the resulting C: \BATCH \CWD.BAT file would be

```
D:
CD D:\WS4\LETTERS
```

At any time, even after powering off or resetting your system, you may return to your current working directory by typing

CWD

Figure C-22 lists the SET-CWD batch job stream, which creates two temporary files in the current directory. After combining them to create C: \BATCH \CWD.BAT, the job stream deletes them. The resulting CWD.BAT is within the program search path, so it may be run from any directory in the system.

Figure C-22. The SET-CWD.BAT job stream defines the current working directory.

```
echo off
REM define the current working directory
cd > mydir.tmp
debug< c:\batch\set-cwd.inp > nul
copy mydisk.tmp/b+mydir.tmp/b c:\batch\cwd.bat/b > nul
del mydisk.tmp
del mydir.tmp
```

SET-CWD, in Figure C-22, uses DEBUG to alter the directory name within the temporary file MYDIR.TMP. DEBUG is controlled by the input script SET-CWD.INP, listed in Program C-5. This script retains the disk drive identifier, and appends a new line containing a CD (Change Directory) command followed by a space.

Program C-5. The SET-CWD.INP input script directs DEBUG to create MYDISK.TMP for the SET-CWD job stream.

```
N MYDIR.TMP
L
N MYDISK.TMP
F 0102 L 0005 0D 0A 43 44 20
R CX
0007
W
Q
```

Appendix D

Sources

Sources

Public Domain Programs

Public domain programs are available at no charge. You may copy and distribute them freely. You may not sell them, nor may you charge for their use. Some authors prohibit using their public domain software commercially, such as in support of a business.

Public domain programs can be a great bargain, but they also have a potential limitation. Because no one is responsible for maintaining them, you may have nowhere to go for help or redress when you encounter problems. While some public domain software is superior to many commercial products, most of it is incomplete and intolerant of errors. Some "trojan horses" are deliberately destructive programs that masquerade as useful tools. Stick to public-domain programs that have been recommended by people whose judgement you have learned to trust; let the other guy separate the wheat from the chaff.

You may obtain public domain programs from

- Friends
- User groups
- Bulletin boards

User groups are an excellent source of information and contacts as well as public domain software. No longer collections of hackers and kids, user groups have grown up. For example, the Capital PC group has 5,000 members with an average age in the mid-40s and and average annual income of $61,000.

Large commercial data services like The Source and CompuServe offer many trustworthy downloadable programs. Many computer stores' user groups offer collections of public domain software at low prices, roughly $5 per disk. One such source is MicroCom Systems, 3673 Enochs St., Santa Clara, CA 95051.

Public domain programs discussed in this book are:

CED
Chris Dunford
Cove Software Group
10057-2 Windstream Dr.
Columbia, MD 21044

KBFIX2 by Skip Gilbrech

KeyBuff

MARK/RELEASE from TurboPower Software

PCopy from Norm Patriquin

ScrnSave

SD

XEQ
Hardwood Software Associates
143 Ash St.
Hopkinton, MA 01748

Shareware

Shareware programs are a hybrid. You can copy them and try them without charge. If you like them, you're asked to send a payment to the author. Often the payment will bring a manual, updates, and additional information. Sometimes it just keeps the author interested in providing more shareware. Increasingly, corporations are turning to shareware for software duplication without troublesome site licensing or potential litigation. The shareware system works well if you don't abuse it.

Shareware programs used while writing this book are:

ARC
System Enhancements Associates
21 New St.
Wayne, NJ 07470
$50

Automenu
Marshall Magee
Magee Enterprises
6577 Peachtree Industrial Bvd.
Norcross, GA 30092

Disk Organizer
Soft GAMs Software
G. Allen Morris III
1411 10th Ave.
Oakland, CA 94606
$20 license fee requested

PSLIST
Controlled Information Environments
P. O. Box 457
La Mesa, CA 92044-0080
$15 per system, $150 per network

Commercial Software

You can usually buy commercial software directly from the manufacturer. Other sources, in decreasing order of both cost and service, are:

- Your trusted computer consultant
- Your local computer or software store
- A mail-order house

If you work with computer consultants on a regular basis, they can recommend software, obtain it, install it, and train you in how to use it. Of course, you pay for the service. Consultants offer experience, knowledge, and support. They do not discount.

You often pay full price at a computer store. Some computer stores discount, but then they only deal with customers who know what they want. Other than aligning disk drives, no one seems to repair computer components any more; they just ship boards back to manufacturers.

Software-oriented stores frequently discount, sometimes heavily. Egghead Software seems to offer exceptional service along with good discounts. Egghead has an excellent corporate buying program. Generally you shouldn't expect a great deal of vendor support, even when you buy locally. Unless you're buying a best seller or you happen to get lucky, there's probably no one in the store who knows any more about the package than you do. If you find you are educating the salesperson, consider using mail order instead.

Because buying locally is no guarantee of support, it often makes more sense to buy by mail. Mail order sources discount heavily, and you can shop around. Pick a reputable firm with an established track record. Computer magazines carry many ads; look at current and back issues to pick a firm whose ads have appeared for at least a year. Increasing ad sizes indicate a growing customer base; shrinking ads mean an eroding market share. You can figure that customer satisfaction is at least part of the reason for either phenomenon, so go with a winner.

The Federal Trade Commission regulates mail-order houses, so you have some protection. Under FTC rules, the seller must ship within 30 days or let you know if there will be a delay. If so, you have the option to cancel without penalty. Even if you agree to wait, you can change your mind and cancel at any time. If you choose to cancel, your money must be refunded within seven days. Credit card charges must be refunded within one billing cycle.

There is always the chance of a snafu when ordering by mail. Check everything carefully and take good notes. Verify the price, availability, the exact merchandise being shipped, and the shipping method and costs. Write down the salesperson's name and the date and time you placed the order. Find out about warranties and where you go for help or service. If you have any written correspondence, keep copies. Do not send cash through the mail.

Commercial programs examined during the preparation of this book are:

BackTrack
Tallgrass Technologies
11100 W. 82nd St.
Overland Park, KS 66214
(401) 274-0393
$129

Command Plus
ESP Software Systems
11965 Venice Blvd. #309Los Angeles, CA 90066
(800) 992-4ESP

Copy II PC
Central Point Systems
9700 SW Capitol Hwy., #100
Portland, OR 97219
(503) 244-5782
$39

CoreFast
Core International
7171 North Federal Hwy.
Boca Raton, FL 33444
(305) 997-6055
$149

Cruise Control
Revolution Software
715 Route 10 East
Randolph, NJ 07869
(201) 366-4445
$29.95

Cubit
SoftLogic Solutions
1 Perimeter Rd.
Manchester, NH 03103
(800) 272-9900
(603) 627-9900 in New Hampshire
$49.95

DataCare
Ellicott Software
3777 Plum Hill Ct.
Ellicott City, MD 21043
(301) 465-2690
$99

Dayflo Tracker
Dayflo Software
17791 Mitchell Ave. N.
Irvine, CA 92714
(714) 474-1364
$99.95

Direct Access
Delta Technology International
P.O. Box 1104
Eau Claire, WI 54702
(715) 832-0958
$89.95

Direc-Tree Plus
Micro-Z
4 Santa Bella Rd.
Rolling Hills, CA 90274
$49.50

dirWORKS
Keep It Simple Software
580 5th Ave.
New York, NY 10036
(212) 398-1286
$25

Disk Engineer
Prime Solutions
1940 Garnet Ave.
San Diego, CA 92109
(619) 274-5000

Disk Optimizer
SoftLogic Solutions
1 Perimeter Rd.
Manchester, NH 03103
(800) 272-9900
(603) 627-9900 in New Hampshire
$49.95

Disk Technician
Prime Solutions
1940 Garnet Ave.
San Diego, CA 92109
(619) 274-5000
$99

DOS2ools
E-X-E Software Systems
8855 Atlanta #298
Huntington Beach, CA 92646
(714) 662-2535
$99

DoubleDOS
SoftLogic Solutions
1 Perimeter Rd.
Manchester, NH 03103
(800) 272-9900
(603) 627-9900 in New Hampshire

DPath Plus
Pesonal Business Solutions
P.O. Box 757
Frederick, MD 21701
(301) 865-3376
$49

Dragnet
Access Softek
Berkeley, CA
(800) 222-4020
$145

DS Optimize
DS Recover
DS Backup Plus
Design Software
1275 W. Roosevelt Rd.
West Chicago, IL 60185
(800) 231-3088
DSBackup Plus: $79.95
DS Optimize: $69.95
DS Recover: $49.95

EasyPath
Polygon Software
330 Seventh Ave.
New York, NY 10001
(212) 967-2424
$100

Electra-Find
O'Neill Software
P.O. Box 26111
San Francisco, CA 94126

Fansi-Console
Hersey Micro Consulting
P.O. Box 8276
Ann Arbor, MI 48107
$25 for software only
$75 for software and manual

Fastback
Fifth Generation Systems
909 Electric Ave. #308
Seal Beach, CA 90740
(800) 228-6127
$179

Fast Forward
Mark Williams
1430 W. Wrightwood
Chicago, IL 60614
(800) MWC-1700
$69.95

File Facility
IBM Personally Developed Software
P.O. Box 3280
Wallingford, CT 06494
(800) IBM-PCSW
$19.95

FilePath
SDA Associates
P.O. Box 36152
San Jose, CA 95158
(408) 281-7747
$37.50

File Recovery System
Brown Bag Software
2105 S. Bascom Ave.
Campbell, CA 95008
(408) 559-4545
$69.95

Flash
Software Masters
6223 Carrollton Ave.
Indianapolis, IN 46220
(800) 25-FLASH
$69.95

Flash-Up Windows
Software Bottling
6600 Long Island Exwy.
Maspeth, NY 11378
(800) 872-8787
$90

FYI 3000 Plus
Software Marketing Associates
4615 W. Bee Caves Rd.
Austin, TX 78746
(512) 327-2882
$395

Hard Desk
Modtech
10933 Crabapple Rd.
Roswell, GA 30075
(800) 223-6250
$95

Homebase
Brown Bag Software
2155 S. Bascom Ave.
Campbell, CA 95008
(800) 523-0764
(800) 323-5335 in California
$89.95

Hot!
Executive Systems
15300 Ventura Blvd. #305
Sherman Oaks, CA 91403
(800) 634-5545(800) 551-5353 in California
$75

HQ
TEK Microsystems
2067 Massachusetts Ave.
Cambridge, MA 02140
(617) 497-1200
$79

HTest/HFormat
Paul Mace Software
123 N. First St.
Ashland, OR 97520
(800) 523-0258
(503) 488-0224
$89.95

IN.SIGHT
Pearlsoft
P.O. Box 638
Wilsonville, OR 97070
(503) 682-3636
$95

Intelligent-Backup
Sterling Software
202 E. Airport Dr. #280
San Bernadino, CA 92408
(714) 889-0226
$149.95

JRAM Combo Pack
Tall Tree Systems
P.O. Box 50690
Palo Alto, CA 94303
(415) 493-1980
$60

Jet
Tall Tree Systems
1120 San Antonio Rd.
Palo Alto, CA 94303
(415) 964-1980
$60

KeepTrack Plus
The Finot Group
2390 El Cmino Real #3
Palo Alto, CA 94306
(800) 628-2828
$79

Keyworks
Alpha Software
30 B Street
Burlington, MA 01803
(617) 229-2924
$89.95

Lightning
Portable Computer Support Group
11035 Harry Hines Blvd. #206
Dallas, TX 75229
(214) 351-0564
$49.95 (copy-protected)
$89.95 (not copy-protected)

Mace Utilities
Paul Mace Software
123 N. First St.
Ashland, OR 97520
(800) 523-0258
(503) 488-0224
$99

Metro
Lotus Development Corp.
55 Cambridge Pkwy.
Cambridge, MA 02142
(800) 345-1043
$85

NoBlink/Accelerator
Nostradamus
3191 South Valley St. #252
Salt Lake City, UT 84109
(801) 487-9662
$49.95

Norton Commander
Peter Norton Computing
2210 Wilshire Blvd #186D
Santa Monica, CA 90403
(213) 826-8032
(213) 453-2361
$75

Norton Utilities
Peter Norton Computing
2210 Wilshire Blvd. #186D
Santa Monica, CA 90403
(213) 453-2361
$99.95

1 Dir Plus
Bourbaki
615 W. Hays
Boise, ID 83702
(208) 342-5849
$95

PathMinder
Westlake Data
P.O. Box 1711
Austin, TX 78767
(512) 474-4666
$39.95

PC Easy II
Transec Systems
220 Congress Park Dr. suite 200
Delray Beach, FL 33445
(800) 423-0772
$85

PCED
Cove Software Group
10057-2 Windstream Dr.
Columbia, MD 21044
(301) 992-9371

PC Tools
Central Point Software
9700 SW Capitol Hwy. #100
Portland, OR 97219
(503) 244-5782
$39.95

PDisk
Phoenix Technologies Ltd.
320 Norwood Park S.
Norwood, MA 02062
(617) 762-5030
$145

Polaris Rescue
Polaris Software
613 W. Valley Pkwy. #323
Escondido, CA 92025
(619) 743-7800
$149

PopDrop
InfoStructures
P.O. Box 32617
Tucson, AZ 85751
(602) 299-5962
$19.95

Power Tools
MLI Microsystems
Framingham, MA
(617) 879-2000
$55

PrintQ
Software Directions
1572 Sussex Turnpike
Randolph, NJ 07869
1-800-346-7638
$89

Program Director
Power Up! Software
Channelmark
2929 Campus Dr.
San Mateo, CA 94403
(415) 345-5900
$49.95

ProKey
RoseSoft
P.O. Box 45880
Seattle, WA 98145
(206) 282-0454
$129.95

Protec
Sophco
1906 13th St.
Boulder, CO 80306
(303) 444-1542
$195

QDOS II
Gazelle Systems
42 N. University Ave. #10
Provo, UT 84601
(800) 233-0383
(801) 377-1288
$44.95

QUICKCache
MSD
214½ W. Main St.
St. Charles, IL 60174
$49.95

Referee
Persoft
465 Science Dr.
Madison, WI 53711
(608) 273-6000
$69.95

SmartKey II Plus
Software Research Technologies
2130 South Vermont Ave.
Los Angeles, CA 90007
(213) 737-7663
$59.95

SmartNotes
Personics
2352 Main St. Bldg. 2
Concord, MA 01742
(617) 897-1575
$79.95

SmartPath
Software Research Technologies
2130 S. Vermont
Los Angeles, CA 90007
(213) 737-7663$29.95

SNAP
The Mt. Whitney Group
11612 Knott Ave. #G-19
Garden Grove, CA 92641
(800) 992-4992
(800) 624-7355 in California
$99.95

Software Bridge
Systems Compabability
1 E. Wackeer Dr. #1320
Chicago, IL 60601
(312) 329-0700
$149

Software Carousel
SoftLogic Solutions
1 Perimeter Rd.
Manchester, NH 03103
(800) 272-9900
(603) 627-9900 in New Hampshire

SpeedStor
Storage Dimensions
981 University Ave.
Los Gatos, CA 95030
(408) 395-2688
$99

SQZ!
Turner Hall Publishing
10201 Torre Ave.
Cupertino, CA 95014
(408) 253-9607
$79.95

SuperKey
Borland International
4585 Scotts Valley Dr.
Scotts Valley, CA 95066
(800) 255-8008
$69.95

Super PC-Kwik
Multisoft
18220 SW Monte Verdi Blvd.
Beaverton, OR 97007
(503) 642-7108
$79.95

TakeTwo
United Software Security
8133 Leesburg Pike #800Vienna, VA 22180
(800) 892-0007
$115

Tallscreen
Qualitas
8314 Thoreau Dr.
Bethesda, MD 20817
(301) 469-8848
$49.95

TopDOS
Frontrunner Development
14656 Oxnard St.
Van Nuys, CA 91411
$69.95

UNLock
Transec Systems
220 Congress Park Dr., suite 200
Delray Beach, FL 33445
(800) 423-0772

VOpt
Golden Bow Systems
2870 Fifth Ave. #201
San Diego, CA 92103
(619) 298-9349
$49.95

Watchdog
Fischer Innis Systems
4175 Merchaantile Ave.
Naples, FL 33942
(800) 237-4510
(813) 793-4150
$295

WordPerfect Library
WordPerfect
288 W. Center St.
Orem, UT 84057
(801) 225-5010
$129

X-Tree
Executive Systems
15300 Ventura Blvd. #305
Sherman Oaks, CA 91403
(800) 634-5545
(800) 551-5353 in California

Zoo Keeper
Polaris Software
623 W. Valley Pkwy. #323
Escondido, CA 92025
(619) 743-7800
$75

ZyINDEX
ZyLAB
233 E. Erie St.
Chicago, IL 60611
(312) 642-2201
$145

Hardware Manufacturers and Sources

You can purchase hardware from consultants, local stores, or mail order houses. Here are manufacturers for products discussed in this book:

Bernoulli Box/Bernoulli Beta 20 half-height internal 20 Mb drive
Iomega
1821 W. 4000 South
Roy, UT 84067
(801) 778-1000
$2,200: Bernoulli Box
$2,600: Bernoulli Beta 20

Cauzin SoftStrip Reader
Cauzin Systems
835 S. Main
Waterbury, CT 06706
(203) 573-0150

Core HC150
Core International
7171 N. Federal Hwy.
Boca Raton, FL 33431
(305) 997-6055
$4,995

DataSaver 400
Cuesta Systems
3440 Roberto Ct.
San Luis Obispo, CA 93401
(805) 541-4160
$695

Diskit 2 Plus dual 10 Mb removable drives with hardware encryption/decryption
IDEAssociates
35 Dunham Rd.
Billerica, MA 01821
(800) 257-5027
$3,595

Emulex ATS-380 310Mb drive with 60Mb cartridge tape backup and EDSI interface
Emulex
3545 Harbor Blvd.
Costa Mesa, CA 92626
(714) 662-5600

525WC and HD525 WORM drives: 230Mb and 1 Gigabyte
Information Storage
2768 Janitell Rd.
Colorado Springs, CO 80906
(303) 579-5222
$2,795 and $4,495

Keyboard slideaway, console tilt/swivel, copy stand
Kensington Microware
251 Park Ave. S.
New York, NY 10010
(212) 475-5200

Personal Data Pac 30 Mb removable Winchester drive
Tandon
805 Science Dr.
Moorpark, CA 93021
(805) 378-6081
$350

Sysgen DuraPak
47853 Warm Springs Blvd.
Fremont, CA 94539
(800) 821-2151
$1295

Verbatim 12 Mb removable half-height internal drive
Eastman Kodak, Mass-Memory Division
343 State St.
Rochester, NY 14650
(800) 445-6325
$1,400

Vertical system stands
Curtis Manufacturing
305 Union St.
Peterborough, NH 03458
(603) 924-3823
(619) 457-5500
$29.95

X2C Expanded Memory Board
ABM/Franklin Computer Systems
733 Lakefield Rd.
Westlake Village, CA 91361
2Mb board: $620

Business mail-order houses and forms suppliers are good sources for hardware accessories. Here are some names and addresses:

INMAC
2465 Augustine Dr.
Santa Clara, CA 95054

Misco
1 Misco Plaza
Holmdel, NJ 07733
(800) 631-2227

Moore Computer Supplies
P.O. Box 20
Wheeling, IL 60090
(800) 323-6230

Appendix E

Glossary

Glossary

Abort
To stop a program or function before it comes to its natural completion.

Accelerator board
An add-on board that uses a faster system clock, faster processor, coprocessor, or other hardware to speed up overall system throughput.

Alignment
Accurate line-up between disk tracks and read-write heads.

Anchor bolt
A hardware protection device which physically bolts a computer system or other hardware to a desk or other immovable furniture.

ANSI.SYS
A device driver, included with MS-DOS, that controls screen colors, graphics, and keyboard output.

APPEND
MS-DOS command (versions 3.3 and higher) to extend the search path for nonexecutable files.

Archive bit
Part of the directory information MS-DOS keeps for each file. The bit indicates whether the file was added or changed since the last backup.

Arm
Part of the disk drive unit which supports a read-write head. There are as many arms as there are read-write heads.

ASCII
American Standard Code for Information Interchange. A standard seven-bit format for data recording, widely used on microcomputers.

ASCII file
A file containing only printable characters, or characters that can be entered directly from the keyboard.

ASSIGN
MS-DOS command to reroute activity from one disk drive to another.

Attributes
Four status characteristics of a file which MS-DOS keeps track of using attribute bits in the directory entry for a file. The four attributes are Hidden, System, Read Only, and Archive.

Audit trail
For security purposes, a log of system activity: It keeps track of who used what files when.

AUTOEXEC.BAT
An ASCII text file which, if located on the root directory of the boot drive, contains startup instructions executed automatically whenever the system boots.

Average access time
The average time to move to a sector and read it from disk into memory.

Average seek time
Average time required for a typical seek.

BACKUP
MS-DOS command to copy a file or group of files. The resulting output is unreadable except with the MS-DOS RESTORE command, but it offers a way to copy more files than can fit on a single floppy disk and to span files across more than one floppy disk.

Bad sector
A sector with read or write errors. MS-DOS marks the sector as bad and does not attempt to use it for data storage.

Batch file
A sequence of MS-DOS commands and control information stored in an ASCII text file. The filename must have the extension .BAT.

Bay, disk
A physical area in a microcomputer reserved for floppy or hard disk drives.

Binary file
A program file or other file containing machine language or other nontext data not readily readable by humans.

BIOS
Basic Input/Output System. The portion of the operating system which controls data transfers between RAM and the keyboard, screen, and storage devices.

Boot
To start a computer system.

Boot record
The first sector of a bootable disk partition. Contains startup instructions to load the operating system and start it functioning.

Buffer
A temporary storage area. A disk buffer is a temporary storage area in RAM used to hold recently read or written disk data. If another request comes for data which is already in the buffer, a disk read can be avoided, speeding processing greatly.

Bus
A set of parallel wires which transfers data between RAM and the keyboard, console, and storage devices.

Cartridge tape
A special tape device, originally developed by 3M Corporation, in which tape inside a specially designed plastic cartridge on a rigid aluminum base is used as the data storage medium.

CD
Abbreviation for the MS-DOS command CHDIR (change directory).

CHDIR
The MS-DOS change directory command. CHDIR selects the current, or default, directory.

Child directory
A subdirectory. If A is a directory, and B is a subdirectory of A, then B is a child of A.

CHKDSK
MS-DOS command to check for bad sectors and restore lost clusters.

Clicking
Using a mouse to position the pointer over an item, then pressing a button on the mouse to indicate selection of the item, simulating the same selection process on the keyboard.

Clipboard
A temporary storage area used in moving or copying data between files.

Clock crystal
An oscillator crystal whose frequency sets the overall pace of system operations.

Clock speed
The frequency of the crystal in the system clock.

Cluster
The smallest unit of disk space MS-DOS will work with in the file allocation table. A typical hard disk cluster size is four sectors.

COMMAND.COM
The MS-DOS command processor. Normally located on the root directory of the boot drive.

Compression
Any of a number of techniques to reduce the space required to store data, typically by using special coding to remove redundancies and irrelevancies.

Concurrency
Running more than one program during the same time period. Concurrency on microcomputers involves switching between the programs so rapidly that they both appear to be processing at once.

CONFIG.SYS
An ASCII text file containing MS-DOS configuration data. Includes such items as the number of buffers, maximum number of files open at once, highest disk drive, and pointers to all device drivers to be loaded by MS-DOS. If CONFIG.SYS is not on the root directory of the boot drive, default settings are used.

Corrupted file
A file whose contents have been damaged and are erroneous. The damage can come from physical causes such as a head crash or from logical causes such as accidentally writing into the middle of the file.

Cut-and-paste
A technique for moving or copying data between files.

Cylinder
All tracks with the same number on all surfaces of a disk. For example, the set of all track threes on a drive is cylinder three. The read-write heads always position over a cylinder: Each read-write head is over the same track number but on a different surface.

Cylinder zero
The outside cylinder on a disk drive, containing the most important information for the drive: master partition table, partition bootstrap, file allocation tables, and root directory.

Data file
Any file which is not a program-related file. Data files are typically user-created and contain the material the application program works with. Examples of data files are documents, spreadsheets, and database files.

Decryption
The process of decoding an encrypted file.

Defragger
A program to reunite files which have split into multiple small, noncontiguous pieces. The end result of running a defragger is a disk in which all files are whole and there are no gaps between files.

Defragment
To run a defragging program.

Desktop organizer
Any of a number of memory-resident programs providing services comparable to the typical office desktop: clock, calendar, appointment scheduler, notepad, telephone list, or calculator, for example.

Device driver
A program that assists MS-DOS in working with a particular input/output or storage device. Device drivers are most often used with nonstandard devices, such as disks larger than the 32-megabyte limit of older versions of MS-DOS.

Directory
A special file whose contents are a list of files and pointers to other directories. Each directory entry lists a filename, last update data, system attributes, and a pointer to the file's starting sector in the File Allocation Table.

Disk cacher
A program to trap and store data as it comes to or from a disk. The program keeps as much data as possible in memory, using a complicated algorithm to determine what data to keep and what to release. The program also traps requests for data reads and writes. If possible, it satisfies read/write requests without going to the disk, thus saving time.

Disk controller
The electronic operating system for a disk drive. The controller handles requests for data, translating them and issuing commands to the read/write heads to perform the required operations.

Disk-on-a-card
A compact hard disk drive, usually packed with its own disk controller, on a plug-in card which fits into a slot on a computer's motherboard.

DOS command editor
Any program providing common text editing features on MS-DOS commands: scrolling, copying, adding and correcting, and combining several commands into one.

DOS shell
A program that supplements MS-DOS by simplifying or improving common MS-DOS functions such as viewing, copying, moving, deleting, or renaming files or directories.

Drive identifier copy-protection
A copy-protection method using a hidden file containing drive-specific identifier information. The protected program will not run unless the information on the hidden file matches the actual status of the drive.

EEMS memory
Extended Expanded Memory Specification memory. A particular method of using otherwise unaddressable memory above the 640K limit. Developed jointly by the Quadram and AST computer accessory manufacturing firms, EEMS swaps 64K memory pages into and out of a memory window. An alternative system, LIM (or EMS) memory, uses 16K memory pages (see LIM memory).

EMS memory
Expanded Memory Specification memory (see LIM memory). Random access memory (RAM) above the 640K limit which matches the paging and other requirements of the LIM specification established by Lotus, Intel, and Microsoft.

Encryption
Any of a number of methods of encoding a file so its contents will appear garbled to anyone without the right password.

Expanded memory
EMS or LIM memory; memory above 640K accessed using a bank-switching or paging technique established by Lotus, Intel, and Microsoft.

Extended memory
Memory above 640K. Also called protected memory.

Extension
The part of a filename after a period. In the example COMMAND.COM, *COM* is the extension. Extensions can be no more than three characters long and follow a period that separates them from the filename.

External disk
A disk drive outside the main body of a microcomputer, with its own casing and power supply.

FAT
See File allocation table.

FDISK
An MS-DOS utility to create or modify an MS-DOS partition on a disk.

File allocation table (FAT)
An MS-DOS table which records the status of every sector on a disk. Each sector is either available, bad (and therefore unusable), or part of a file (unavailable). If a sector is part of a file, its FAT entry contains a pointer to the next sector in the file.

File-by-file backup
A backup in which each file is copied as a separate entity and can be restored individually.

Footprint
The physical space a system occupies on a desktop or floor.

Format
To prepare a disk, either (1) low-level physical preparation of sectors, tracks, cylinders, and the identifying markers for each, or (2) logical formatting establishing the root directory and file allocation table.

FORMAT
MS-DOS command to perform logical formatting on a disk (see format, above).

Fragmented disk
A disk containing noncontiguous files. Noncontiguous files are files split into two or more physically separate pieces on the disk.

Half-height
A disk drive filling only one-half the height of a disk bay. Two half-height drives can fit into a single disk bay.

Hardware key copy-protection
A copy-protection method using a separate small hardware device, attached to the computer internally or through a cable, which communicates with the protected software.

Head crash
An accident in which a disk head makes physical contact with a disk surface, damaging the disk surface's magnetic recording medium.

Head, disk
The electromagnetic unit that reads and writes data on a disk.

Hidden files
Files with the MS-DOS hidden attribute. These files will not appear in an ordinary directory listing, and are in essence invisible to any user not sophisticated enough to look for their presence.

Image backup
An exact copy of the data on a disk or other storage device, without regard to contents or organization. Usually such a backup can only be restored in its entirety. It is extremely difficult to locate and restore individual files.

Incremental backup
A backup of all files which were added or changed since the last backup, as indicated by the MS-DOS Archive bits in the directory entries.

Interleave
The number of sectors separating two logically adjacent sectors on a disk. An interleave of 1:1 means the sectors are next to each other: 1,2,3. . . . An interleave of 2:1 means one sector separates two logically adjacent sectors: 1,4,2,5,3. . . .

Internal disk
A disk within the case of a microcomputer, using the computer's power supply and controlled by a card attached to the computer's bus.

I/O
Input/output. Anything involving moving data between computer memory and external devices such as the keyboard, console, printer, or disk drives.

Job stream
A related series of computer tasks involving a number of programs and/or MS-DOS commands. Batch files, macros, and MS-DOS command synonyms are ways to create a single command that initiates a job stream.

JOIN
MS-DOS command (version 3 and higher) to let MS-DOS regard a disk drive as a subdirectory of another drive.

Key disk copy-protection
A method of copy-protection that requires putting the original, specially marked program disk in drive A.

Latency
Time needed for one sector to spin under the read-write heads.

Level-one directory
A directory whose parent is the root directory.

LIM memory
A particular method of using otherwise unaddressable memory above the 640K limit. LIM stands for Lotus, Intel, and Microsoft—the three companies who developed the LIM standard. The LIM specification swaps 16K memory pages into and out of a 64K window. An alternate upward-compatible specification, EEMS, uses a memory page size of 64K.

Line conditioner
A power-line enhancement that flattens irregularities in line voltage, reducing the danger of damage from spikes, surges, or brownouts.

Locking keyboard cover
A hardware protection device that physically covers a keyboard, disallowing keyboard entry until the cover is removed.

Lost clusters
Clusters marked as in use which are not part of any file. The MS-DOS CHKDSK command finds and recovers lost clusters.

Macro
A series of commands and pseudo-keyboard input triggered by a single keystroke or small combination of keys.

Macro processor
A program to create, edit, and run macros.

Master boot record
A program on the first sector of cylinder zero, side zero of a bootable hard disk. This program reads the partition table (located in the same sector), identifies the active partition, and loads its first sector, which should contain the bootstrap for the partition's operating system.

MD
An abbreviation for the MS-DOS MKDIR command, which creates a new directory.

Memory
A computer's fast electronic data storage area. Memory usually comes in two forms: random access memory (RAM) and read only memory (ROM).

Menu
A listing of choices that provides a simple way to indicate your selection.

Menu builder
Any program providing tools enabling users to develop custom menus quickly and easily.

Metal-oxide media
An older magnetic recording medium using an epoxy coating on a hard disk surface with embedded magnetic oxide particles.

MKDIR
MS-DOS make directory command, which creates a new directory.

Monitor valet
An accessory that raises a video monitor above the work surface, providing easy adjustment of the position of the monitor.

Mounting rails
Plastic or metal bars installed on the sides of hard disks or floppy disk drives upon which the drives rest when mounted within a computer. This method is used for mounting drives within the PC AT.

MS-DOS
Microsoft Disk Operating System. The main control program for IBM PC–family computers. Versions sold by IBM are called PC-DOS.

Multitasking
Keeping more than one program loaded and ready to run. The programs are not necessarily concurrent. It may be necessary to suspend operations in one program in order to run a second. However, when you return to the first program, you continue from the exact point and conditions which existed at the time of suspension.

Nonvolatile RAM
Random access memory which can retain data even without power to the main computer system. Nonvolatile RAM usually has its own power supply and a battery backup to continue operation even through a power failure.

Optical drive
A disk drive using optical (laser) rather than magnetic technology to store data. Slightly different levels of reflectivity are used to indicate on and off states.

Optimizer
A program that defragments files and/or rearranges files within a disk directory by putting the most-read files first and most-written files last.

Overlay file
A file containing part of a computer program, loaded by the main program file as needed.

Parent directory
The directory in which a subdirectory is created. The parent directory contains a pointer to the subdirectory, or child directory. In the child directory, the .. directory entry points to the parent directory entry.

Parking
The process of moving a disk's read-write heads to a safe, data-free area and locking the heads in place when the machine is turned off. Even if the head touches the media on a parked disk, no data damage results.

Partition
A logical section of disk used by a single operating system. MS-DOS normally allows up to four partitions on a disk, although device drivers may change this limit.

Partition loader
Another name for the master boot record.

Password
A secret word or code identifying a particular user or group of users of a system, disk, directory, or file.

Path
(1) The full set of directory names leading from the root directory of the drive to a file, for instance, *C:\wp\data\prop\chapter2.txt.* (2) The set of directories on which MS-DOS will search for .EXE, .COM, or .BAT files. The command search path is usually set with the PATH command.

PATH
MS-DOS command used to select which directories MS-DOS should search for program files (files ending in .EXE, .COM, and .BAT), or to list the current search path.

Path extender
A program enabling multidirectory searching for nonprogram files, similar to the search capabilities provided by the PATH command for program files.

Platter
One physical disk on a disk drive. One platter usually has two usable surfaces, top and bottom. A hard disk drive may contain one, two, or many platters.

Point and shoot
A selection technique in which you move the cursor or a highlight bar to the desired item, then press Enter or some other key to indicate your selection.

Pop-up
A memory-resident program or menu which appears any time you press a predefined key or key sequence.

Print spooler
A program to buffer, control, and monitor data going to a printer.

Program file
An executable file with a .EXE or .COM extension.

Program support file
A subsidiary file used by a program file. Usually these are read-only or rarely written files supplied by the manufacturer as an integral part of an application program. Help files, control files, and spelling dictionaries are examples of program support files.

RAM
Random access memory. See Memory.

Ramdisk
Also called virtual disk; a pseudo-disk using RAM as the storage medium. Also called an electronic disk. A ramdisk is much faster than a true disk. Once created by ramdisk manager software, a ramdisk appears just like a real disk drive to MS-DOS.

RD
Abbreviation for the MS-DOS remove directory (RMDIR) command.

Read only
(1) A file which is only read, never written to.
(2) One of four MS-DOS attribute bits for each file. If the Read-only attribute bit is set, you may not remove or change the file.

Read-write head
The magnetic apparatus which, when positioned over a track, reads and writes data by creating or sensing magnetic flux.

Removable disk
A disk drive whose storage media are removable. The media are protected by a plastic cartridge housing.

RESTORE
MS-DOS command to recopy backup copies made with the BACK-UP command, restoring them to standard MS-DOS readable format.

RLL
Run Length Limited encoding. A data recording technique which squeezes up to 50 percent more data onto a disk. Requires a special RLL disk controller and a high-quality disk. Carries slightly greater potential danger of read/write problems.

RMDIR
MS-DOS remove directory command. Removes a directory by erasing its directory pointer. A directory must be empty for RMDIR to work.

Root directory
The first directory on a disk, and direct ancestor of all other directories on the disk.

Search path
The list of directories to search for a file. Established with the MS-DOS commands PATH or APPEND or with a path-extender utility.

Sector
The smallest logical piece of a disk, typically 512 adjacent bytes. All tracks on a disk contain the same number of sectors, so sectors on outside tracks are physically larger than sectors on inside tracks.

Security board
An add-on board which incorporates several protective features such as hardware encryption/decryption, activity logging, bypass protection, and passwords.

Seek time
Time required to move the read-write head to a sector.

Self-parking
A feature of the disk drive contoller that automatically parks the read/write head on power down.

Servomotor
A motor with a sensing/feedback mechanism. Expensive disk drives usually use a servomotor to move the read-write heads. One disk surface usually contains the location information for the servomotor's feedback mechanism.

Shock mounting
A method of mounting the disk drive intended to protect the disk drive from damage due to sharp movement or vibration. Rubber grommets are the most common form of shock mounting.

Sibling directories
Two directories with the same parent.

Stepper motor
A motor which moves in discreet units or steps. Inexpensive disk drives frequently use stepper motors. To move the read-write heads to a track, a stepper motor starts the heads at cylinder zero, then steps one track at a time until the heads reach the desired destination.

Streaming tape
A fast tape drive using read-while-writing data verification. Streaming tape drives are commonly used as backup devices for microcomputers.

SUBST
MS-DOS command (versions 3 and higher) enabling MS-DOS to treat a subdirectory as though it were a separate disk drive. Some programs, such as *WordStar* version 3.3*x* or earlier, can use data files on separate drives but not in subdirectories.

Surge protector
A power line enhancement that dampens power surges.

System attribute
An MS-DOS attribute bit is associated with each directory entry. The bit is normally on only for files which are an integral part of the operating system, such as IBMBIO.COM.

System clock
The oscillating clock crystal that regulates instruction execution timing.

Tagging
The process of marking related and unrelated files for processing by a fast file manager. Typically, tagged files can then be moved, copied, deleted, printed, or renamed as a group.

Text file
A file consisting primarily of ordinary letters, numbers and characters—readily recognizable text. Normally, an ASCII data file.

Thin-film media
A recently developed magnetic recording medium for disk drives. It consists of an extremely thin layer of plated oxide material which can record data far more densely than the older metal-oxide coatings.

Tilt-and-swivel
A monitor base which lets you rotate or tilt the screen for easier viewing.

Track
One circle of sectors on a single disk surface, all equidistant from the center of the disk. One hard-disk track typically holds 17 sectors. A track is also the area of one disk surface accessible by a read-write head without moving the head.

Track-to-track seek time
The time needed to move from one track to an adjacent track.

Tree-structured directory
A directory that contains files which are themselves directories.

TSR
Terminate-and-Stay-Resident. A type of program that continues to reside in RAM (Random Access Memory) even after it ends processing. TSR, or memory-resident, programs provide many common utility functions such as macro processing, MS-DOS command editing, desktop organizers, file managers, print spoolers, and more.

Type-ahead buffer
A buffer to hold keyboard input. MS-DOS includes a 16-character buffer. Many keyboard utilities provide expanded type-ahead buffers, letting you get farther ahead of DOS.

Unerase
A process of recovering a file or directory which has been deleted with the MS-DOS DELete or ERAse command. Disk/file management utilities such as *Norton Utilities* and *Mace Utilities* provide this function.

Uninstall
The process of removing copy-protected software from your hard disk before any operation, such as defragmenting, optimizing, or file-by-file backups, which could harm the copy-protection program's ability to recognize the file's legitimacy.

Uninterruptible Power Supply (UPS)
A power supply which stays constant even during power outages. A UPS uses a battery pack for power. The battery pack is recharged continuously while outside power is available.

VDISK
MS-DOS utility to create a virtual disk (ramdisk).

Vertical system stand
A device which holds the main system unit of a microcomputer vertical rather than horizontal. Positioning the system unit vertically lets you keep a smaller desktop footprint.

Voice-coil positioner
A motor which relies upon position information recorded on a special surface (the servo track) to move the read/write heads of a disk drive from the current disk track directly to another.

Warm boot
Restarting a computer system without turning off the power, usually by pressing Ctrl-Alt-Del or a reset button.

Well-behaved
Within a windowing environment such as Microsoft *Windows* or IBM's *TopView,* a well-behaved program is one which can successfully operate within a limited window, sharing the screen with other programs and functions.

Whitney technology
A recent improvement in read-write heads and arms, featuring lighter heads which can move more quickly and record more densely than before.

Wildcard
An special character used in an MS-DOS filename which identifies a group of files rather than a single file. The wildcard character ? matches any one character. The character * matches any number of characters.

Winchester drive
A type of disk drive, first developed by IBM, in which the heads and magnetic platters are hermetically sealed.

Window
A rectangular subsection of the screen used for a single purpose.

WORM
Write Once, Read Many times. A type of high-capacity optical disk drive which uses a laser to record data by burning tiny holes into the recording surface.

Wye connector
A cable connector allowing two cables to share a single input.

XCOPY
MS-DOS command (version 3 and higher) to copy groups of files. XCOPY combines the best features of the MS-DOS COPY and BACKUP commands.

Index

Hard Disk Management
Easy Does It

Power users everywhere are discovering the amazing value and performance of the new, inexpensive, bigger-than-ever hard disks for the IBM PCs and compatibles. *Hard Disk Management* makes the hard disk as easy to use as it is fast.

Who should read this book? Anyone who is considering the purchase of a hard disk. Anyone who has recently bought a hard disk. And anyone who is currently using a hard disk and wants to use it more effectively.

Here are just a few of the topics covered in this comprehensive guide:

- Insiders' advice on providing a trouble-free environment.
- Installation and maintenance.
- Security.
- Partitioning and formatting.
- Streamlined operation.
- Data and application backup.

Authors Steven Fisher and Lynn Frantz have years of experience in programming and writing—and in hard disk management. Their complete and readable guide will help you turn your hard disk into your most valuable tool.

$18.95

ISBN 0-87455-116-1
51895
9 780874 551167